K77-27U(R)

GO MANUAL

Wilson Y. Gateley
R. Larry Williams

14 April 1977

KAMAN SCIENCES CORPORATION
A KAMAN COMPANY
1500 Garden of the Gods Road
Colorado Springs, Colorado 80907

This project funded by
Electric Power Research Institute
Palo Alto, CA.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 K77-27U(R)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 GO Manual.		5. TYPE OF REPORT & PERIOD COVERED 9 Final
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) 10 Wilson Y./Gateley [redacted] R. Larry/Williams		8. CONTRACT OR GRANT NUMBER(s) 15 FØ8635-77-R-Ø191
9. PERFORMING ORGANIZATION NAME AND ADDRESS KAMAN SCIENCES CORP. 1500 Garden of the Gods Road Colorado Springs CO 80907		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 64602F 3133
11. CONTROLLING OFFICE NAME AND ADDRESS ADTC/PPR, Eglin AFB FL 32542		12. REPORT DATE 11 24 Apr 77
		13. NUMBER OF PAGES 259 12 274p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) ADTC/SES, Eglin AFB FL 32542		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same as 16.

18. SUPPLEMENTARY NOTES

None.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

GO, safety analysis, reliability analysis, safety reliability, system modeling, GO modeling.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

GO is a modeling procedure for making probabilistic analyses of the success or failure modes of a system. A GO model of a system is created by representing the elements and logical features of a particular system by one of sixteen standard types of components. Probabilities are assigned by the analyst to the operational states of the components, and the GO computer programs compute the probability of the system being in each of its possible

(cont. on reverse side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

389 119

UNCLASSIFIED JOB

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (cont.)

states. The purpose of the GO manual is to explain the rationale of the GO procedures and to provide the detailed instructions for working with the program computers.

K77-27U(R)

GO MANUAL

Wilson Y. Gateley
R. Larry Williiams

14 April 1977

This project funded by
Electric Power Research Institute
Palo Alto, CA.

KAMAN SCIENCES CORPORATION
1500 Garden of the Gods Rd.
Colorado Springs, CO. 80933

ACKNOWLEDGEMENT

The authors acknowledge the counsel and direction received from Drs. Gerald S. Lellouche and Boyer B. Chu of the Electric Power Research Institute in conducting this research and further developing the GO methodology.

GO MANUAL

EXECUTIVE SUMMARY

GO is a procedure for making probabilistic analyses of some of the behavioral characteristics of systems. The term "system" is used here in a broad sense and refers to a collection of basic components (elements) which are connected together and interact in some manner. Sixteen standard types of components are presently defined for use in GO, the types differing from one another in their operational natures. A few component types are perfect in the sense that their outputs are nonstochastic functions of their inputs, but most types represent random devices - that is, devices which may operate in one of several mutually exclusive states.

A GO model of a system is created by representing the elements and logical features of a particular system by suitable GO components which are connected together in a manner which represents the pertinent aspects of the actual operation of the system. Probabilities are assigned by the analyst to the operational states of the components, and the GO computer programs compute the probability of the system being in each of its possible states.

GO is basically an event tree analyser, but by combining certain tree branches at appropriate points and, if necessary, pruning low probability branches, extremely large trees can be successfully handled. The pruning process introduces some error, but the total error is known and by judicious modeling can usually be kept to an acceptably low value.

The events dealt with refer to values of random variables where the random variables are the inputs to and outputs

from the GO components. The random variable values are restricted to a small (up to 128) set of discrete nonnegative integers. For some problems only two values are required (representing, say "good" and "bad"), but usually more than two are used in order to represent relative time values for a system which involves a temporal sequence of operations or when one or more quantifiable attributes are pertinent.

The purpose of the GO Manual is to explain the rationale of the GO procedure and to provide the detailed instructions for working with the computer programs. A large number of examples are included and serve to educate the GO analyst in some of the ways in which the GO components can be used to create a model of a real system. The manual will also be useful as a reference for the experienced user.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF TABLES

LIST OF TABLES (continued)

CHAPTER 1
INTRODUCTION

1.0 BACKGROUND

The purpose of this manual is to introduce the reader to both the theory and application of the GO system analysis procedure. The manual will also serve as a reference for the active GO user.

The goal of a GO analysis is to elicit certain information about the probabilistic behavior of a system consisting of a set of interrelated components whose probabilistic and operational behaviors are specified. The model components themselves are very general in nature and may represent actual physical entities, external actions, conceptual logic devices, etc. Like any specific analytical method, GO does not produce all answers to all questions, and its effective use is highly dependent upon the knowledge and skill of the user. Wisely used, however, it can be a powerful tool for gaining an understanding of many aspects of the nature of a system.

GO was originally created at Kaman Sciences Corporation in 1968. Its development was motivated by the need for a simple yet accurate method for making safety and reliability analyses of complex electro-mechanical systems. Prior to the development of GO these analyses were performed by an equation-writing method which, although exact, was extremely laborious and time-consuming. The use of GO slashed the analysis time from months to days, and the possibility of making numerous "what if" excursions was greatly enhanced by the basic simplicity of the GO models.

Since its inception, GO has been applied to a wide variety of systems, and its versatility and ease of use have

been amply demonstrated in these applications. (A list of many of the specific GO applications is included in Appendix C).

1.1 Using the Manual

For the beginner, a thorough reading of Chapter 2 is mandatory because it is there that the basic GO concept is explained and illustrated.

Chapter 3 contains a brief review of several modeling methods which are in use today and serves to place GO in perspective relative to them. This chapter may be omitted at first, but it should prove to be of some value to an experienced reader.

Chapters 4 through 10 provide the details of creating and exercising a GO model. We suggest a quick first reading of these chapters because a great deal of cross-referencing will be required before it is all satisfactorily digested.

Chapter 11 contains the analyses of a number of specific systems. These examples have been selected to demonstrate as many of the features of GO as possible and should be studied carefully.

Appendix A provides a quick review of the basic probability theory which is required for the understanding of GO.

Appendix B contains the program inputs and outputs for a ficticious system model. This model and the printouts are included to demonstrate many of the mechanical operational features of GO and should be perused from time-to-time.

Appendix C lists a number of projects in which GO has been used in the past.

Appendix D is a glossary of many of the terms used in GO.

Appendix E contains brief summaries of the required operator and kind data, the data deck structures, and the user-defined parameters. These should be useful to the active user as a quick reference.

The serious student is urged to create and run on a computer as many small GO models as possible. Modeling of any kind is an art, and proficiency can be obtained only by practice.

1.2 Related Manuals

The sequel to this "GO Manual" is the "Fault Finder Manual" which covers the theory and use of the Fault Finder programs. These programs are designed to help answer the question of what causes certain events in a system to occur. For example, the GO-modeled system may be a fault tree, and the Fault Finder can then be used to find the minimal cut sets.

The more sophisticated user who is knowledgeable about Fortran may find some interest in both the "GO Program Manual" and the "Fault Finder Program Manual". These manuals document the several computer programs themselves, and although written primarily for maintenance and debugging purposes, provide the specific details of the algorithms and their implementation.

This page intentionally left blank

CHAPTER 2

GO THEORY AND PRACTICE

2.0 INTRODUCTION

In this chapter the basic concepts of GO are discussed. We start with a very simple problem and use it to define and develop most of the important features of the GO modeling method. This is followed by a slightly more complex example for which portions of the actual computer program printouts are displayed. The last section of the chapter discusses the general philosophy of GO modeling.

2.1 The Theory of GO - A Simple Example

At first glance the example given here bears little resemblance to a complicated and largescale system. We urge the reader to study it carefully, however, because it is only in such a simple and "obvious" problem that the basic principles can be exposed with clarity. The GO programs are relatively complex and it is easy to overlook the fundamental simplicity of the GO concept.

Joe has been dithering for some time about whether or not to send a request to his boss for a new desk. To settle the matter he decides that next Friday he will toss a coin. If it comes up heads, he will write a request, otherwise he will forget the whole idea. The boss is a fairly observant and sympathetic individual, and there is a 20% chance that he will send a requisition to purchasing on his own before Friday. If he doesn't, there is a 75% chance he will approve Joe's request if he gets one. To complicate matters, the purchasing department has a rather flighty secretary, and there is a 30% chance that she will lose the requisition if one is sent. Problem: find the probability that purchasing will order a new desk (1) before Friday (2), on Friday, and (3) never.

In order to formalize the problem, let us define three random variables X_1, X_2, and X_3 whose values will be the times at which Joe, his boss, and purchasing respectively act on the desk request or order. Also let us "code" time with the following numerical values:

0: before Friday
1: on Friday
2: never

with these definitions our problem can be restated to read: "Find the probability distribution of the random variable X_3" - that is, find the three missing probabilities in the following table:

Value of X_3	Probability
0	?
1	?
2	?
Total Probability	1.00

TABLE 2.1. THE INCOMPLETE FIND DISTRIBUTION.

A simple method of solution is to construct a probability event tree. Such trees are convenient - at least conceptually in a wide variety of problems involving multiple random variables and particularly when analyzing a sequential process in which dependence between some or all of the random variables exists. The "obvious" tree for our problem is shown in Figure 2.1 and the resulting joint probability distribution of X_1, X_2, and X_3 in Table 2.2. The reader should spend a minute or two verifying the tree construction. Note that the twig probability of .6 which is associated with the boss approving Joe's request is the conditional probability the request is approved given the boss didn't send a requisition on his own multiplied by the probability of the conditioning event - that is, 0.6 = 0.75 (1-0.2).

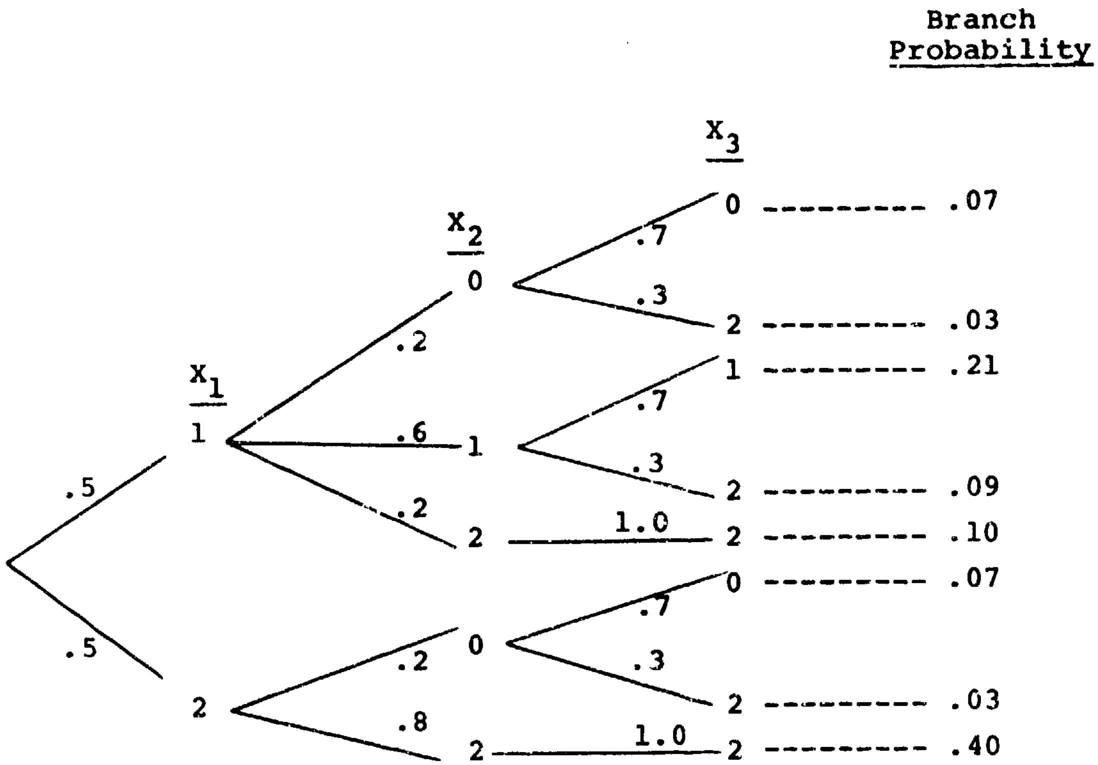


FIGURE 2.1. PROBABILITY TREE.

X_1	X_2	X_3	Probability
1	0	0	.07
1	0	2	.03
1	1	1	.21
1	1	2	.09
1	2	2	.10
2	0	0	.07
2	0	2	.03
2	2	2	.40
			1.00

TABLE 2.2. COMPLETE JOINT PROBABILITY DISTRIBUTION.

From the joint distribution we can immediately obtain the required (marginal) distribution of X_3 by simply summing the appropriate probabilities. We obtain Table 2.3.

Value of X_3	Probability
0	.14
1	.21
2	.65
Total Probability	1.00

TABLE 2.3. THE FINAL DISTRIBUTION.

We note that, if desired, we could also have obtained the joint marginal distribution of, say, X_1 and X_3:

		X_1	
		1	2
X_3	0	.07	.07
	1	.21	.00
	2	.22	.43

TABLE 2.4. JOINT FINAL DISTRIBUTION.

This gives us a little more information about the overall process but at the expense of a little more work.

The "event tree" solution of the New Desk problem exemplifies the basic "theory" of GO. Some changes in the terminology, the introduction of a few concepts, and the use of several techniques to reduce the labor and extend the application scope will see us through to the "practice" of GO. We will continue to refer to the New Desk problem as these additional ideas are discussed.

2.2 Some GO Definitions and Notations

It was mentioned in Chapter 1 that the original application of GO was to analyze electro-mechanical systems. For that work the term signal was used instead of random variable. This usage has been retained in practice, and we shall continue it in this manual. The mathematically oriented reader may replace the word "signal" with "random variable" as he sees fit. A signal will be symbolized by a positive integer. Thus we will speak of "signal 3" rather than "random variable X_3".

The values of the signals (random variables) in a GO model are restricted to a set of nonnegative integers 0, 1,2,...,n where n is some user-defined value less than

128 (in the current version of GO). This restriction is imposed because of the manner by which these values are represented in the computer. It clearly limits the scope of application of GO, but has not been a particularly severe problem in dealing with a wide variety of system analyses.

The value of n, for reasons which will become clearer as we progress, is usually referred to as infinity or never. The latter term has been used in those applications where the signal values represented sequential time points or intervals. These terms are reasonable because n is the largest value that a signal may take. We will normally symbolize n by the symbol ∞ (infinity).

The event that signal S takes on the value i will normally be symbolized by either S_i or $S(i)$, and the probability of the event by $P(S_i)$ or $P(S(i))$.

Basically an event tree represents the construction of a sequence of probability distributions. In general, the "next" distribution is formed by applying a random function to the "current" distribution. This function - which we will refer to as an operator - creates a new random variable (or in some cases several) and forms the joint distribution of the new signal(s) together with the old ones. The tree must, of course, be started with a "constant" operator - that is, a function having no arguments.

In the New Desk problem we have three operators which are identified with the system components, Joe, his boss, and the purchasing office secretary.

We observe that when an operator operates upon the previous distribution, it will frequently require the value of only some of the signals in that distribution in order to generate the new signal distribution. Thus in our example the purchasing office secretary needed only the values of

signal 2 (random variable X_2) in order to generate signal 3 (random variable X_3). We will call the old signals required by a particular operator its input signals and the resulting new signal(s) the output signal(s).

Let us pause to introduce the GO Chart. This is simply a diagram which indicates the relationships among the various operators and signals. Denoting the three operators of our problem by O_1, O_2, and O_3 (for Joe, his boss, and the secretary respectively), we can draw the following diagram where the numbers on the lines coming from an operator are the identification numbers of the output signals:

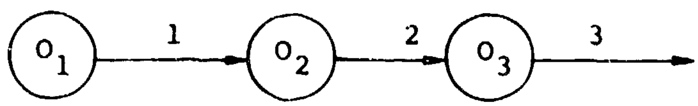


FIGURE 2.2. INITIAL GO CHART

(we will add some additional information and make a few alterations to this diagram shortly). Although a GO chart is not an absolute necessity for making a GO analysis, its use is almost mandatory for a large problem in order to prevent the analyst from becoming hopelessly confused. We note that in many cases the GO chart can be superimposed over a schematic or block diagram of a system. This is a great help to both the analyst and the ultimate user in associating the components of the GO model with the components of the actual system.

At this point it is necessary to define the algorithmic nature of each operator - that is, what are the rules, formulas, etc., by which an operator produces its outputs. It

is convenient to break this definition into two parts, first defining the structure of the operator algorithm and then defining the particular values of the necessary parameters. For example, in elementary mathematics we might refer to the linear function $f(x) = a+bx$ which displays the structure of the function, but to define a specific linear function of this form the values of the parameters a and b must be explicitly given.

It has been found that for the purpose of a GO analysis a small set of different structured forms is adequate. The current version of GO incorporates sixteen of these operator types. Each type is identified by a number (1 through 17 with type 4 being nonexistent). Each type has been selected to provide a reasonable modeling representation of certain physical devices, logical operations, or operational concepts. These types, which constitute the basic building blocks of a GO model, are summarized in Figure 2.3 and are explained in detail in Chapter 6. We will discuss below the three types used in the New Desk problem in order to give the reader some feeling for the concept. The competent GO analyst must, of course, become intimately acquainted with the exact algorithmic nature of all of the operator types. The reader will note that descriptive names have been given to each type. It must be emphasized that these names are suggestive only and are not definitive. For example, a type 2 operator is frequently called an "OR Gate". In some applications it does indeed function like a logical OR gate, but it is fundamentally a more general concept.

The collection of specific parameters for a given operator is referred to as the kind data. The kind data required depends upon the particular type (some types do not require any).

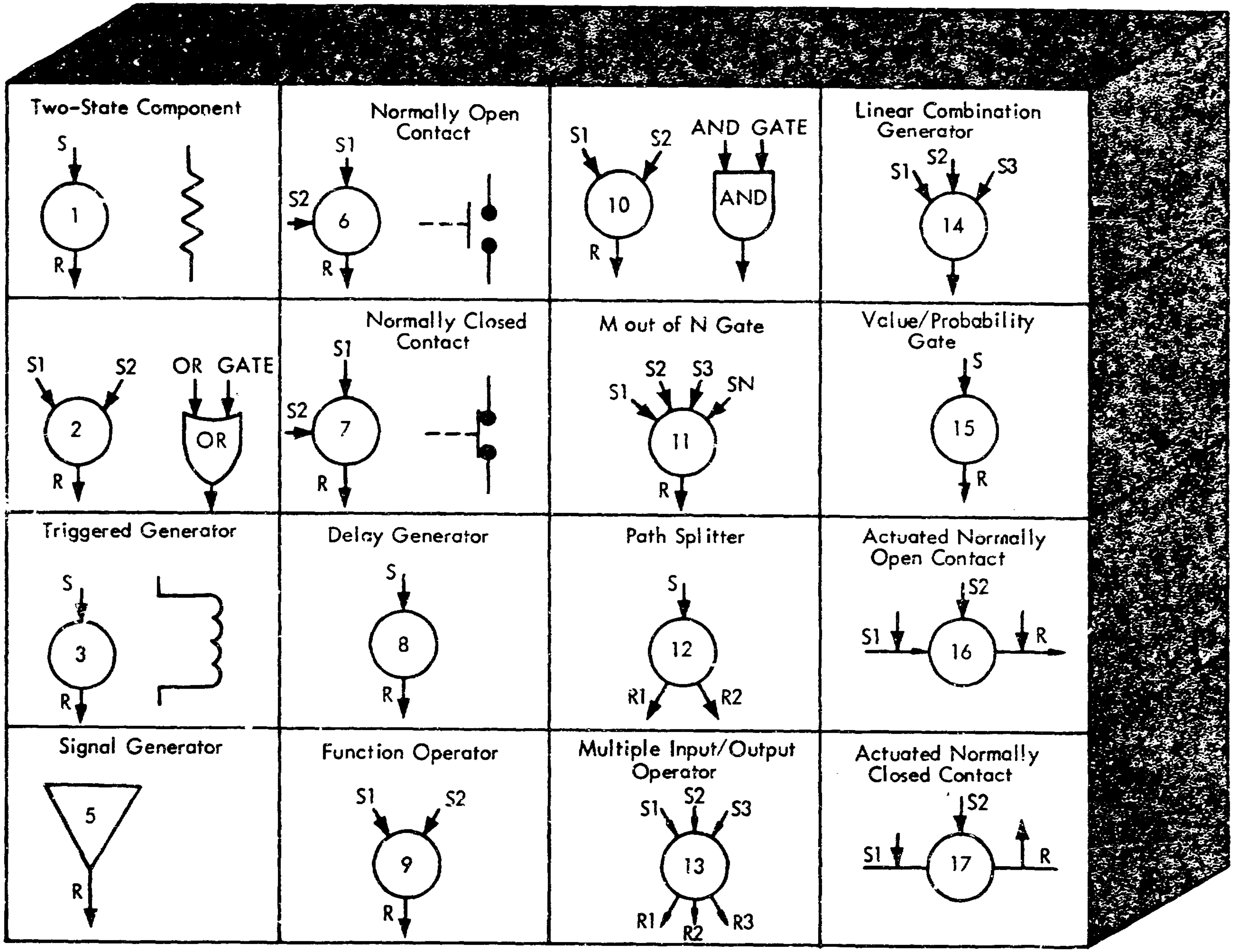


FIGURE 2.3. GO BUILDING BLOCKS.

Within a single model there may be several operators of, say type 1, and each of these may, if desired, be of different kinds - that is, have sets of different parameter values. On the other hand, two operators of different types cannot be of the same kind. In general, the kind data will consist primarily of probabilities and signal values.

2.3 GO Model of The Example

In our example we will model Joe (O_1), with a type 5 operator, his boss (O_2) with a type 3, and the secretary (O_3) with a type 1. Each of these types requires kind data and we will identify these kinds as 10, 20, and 30 respectively (the kind identification numbers are assigned by the modeler and are, within certain restrictions, arbitrary).

The type and kind of each operator are normally indicated on the GO chart. Also, it is usually convenient to indicate a type 5 operator - which represents a starting point or a signal generated externally - by a triangle rather than a circle.

The operator numbers themselves are seldom shown on the original GO chart because they are assigned by the GO program according to the sequence in which the operator data records are given to the computer. In nonserial systems, there will be more than one permissable sequence, and sometimes several iterations may be needed in order to find the best one (in terms of ultimate accuracy and minimum computer time).

With the above comments in mind, the GO chart of our example can be redrawn as shown in Figure 2.4. Note that within an operator symbol the type and kind numbers are shown, separated by a dash.

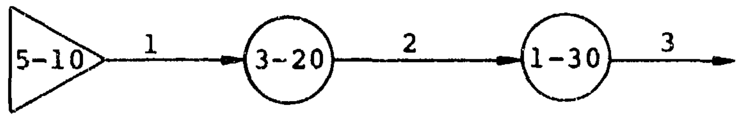


FIGURE 2.4. FINAL GO CHART.

The two tables below show the operator and kind data in essentially the form required for input to the GO computer programs. We note that with the exception of a few additional cards, these are the only data required and hence constitute the GO model as far as the computer is concerned.

Type	Kind	Signals Input	Signals Output
5	10	None	1
3	20	1	2
1	30	2	3

TABLE 2.5. EXAMPLE OPERATOR DATA.

Kind	Type	Parameters (explained below)
10	5	2 1 .5 2 .5
20	3	.6 .2 .2
30	1	.7 .3

TABLE 2.6. EXAMPLE KIND DATA.

The operator and kind data contains the information needed by the computer to create the event tree in the manner described below. Note that the computational algorithms for the various operator types are written in the programs and are user-controlled only to the extent that they are dependent upon the user-supplied kind data.

The first operator - a type 5 (Signal Generator) - is controlled by the kind 10 data to produce a signal with 2 values. These values are to be 1, with probability 0.5, and 2, also with probability 0.5.

The second operator - a type 3 (Triggered Generator) - has three intrinsic operational modes: "good", in which the value of the input signal is simply passed through; "failed" (dud), in which the output value is infinite regardless of the input signal value; and "premature", in which the output value is 0 regardless of the input. The probabilities of the operator acting in these modes are the parameters in the kind data (in the same respective order).

The third operator - a type 1 (two-state component) - has two intrinsic operational modes: "good" and "failed" which perform like the same modes in the type 3 (in fact, we could have used a type 3 with the premature probability set to zero).

The reader should have little difficulty verifying that these three operators do indeed produce the tree in Figure 2.1.

2.4 Special GO Technique

We now turn to the problem of keeping the tree size down to a manageable size. Clearly for a problem in which there are several hundred operators the number of branches in the tree would rapidly become excessive even for a very large computer. GO uses two methods, branch combination and pruning to keep the tree size manageable.

The branch combination technique is based upon the observation that in very few problems are we interested in the joint distribution of all of the signals. This, together with the fact that a given signal is usually used

as an input for at most only a few operators, suggests that we might in essence retain at any stage just the joint marginal distribution of those signals which are needed either as inputs to later operators or to answer the questions the analyst is asking about the system (we will call the latter set the <u>final signals</u>). Thus, in our example, we readily see that once operator 2 has formed signal 2, signal 1 is no longer needed because it is not a final signal (our interest was in signal 3 only) and operator 3 needs only signal 2 to perform its function. Consequently, after the joint distribution of signals 1 and 2 has been formed by operator 2, we can "integrate out" signal 1 and work with just the marginal distribution of signal 2. If we think of the process of "adding" signal 2 and "deleting" signal 1 as occurring simultaneously, we can redraw the tree of Figure 2.1 and get the form shown in Figure 2.5.

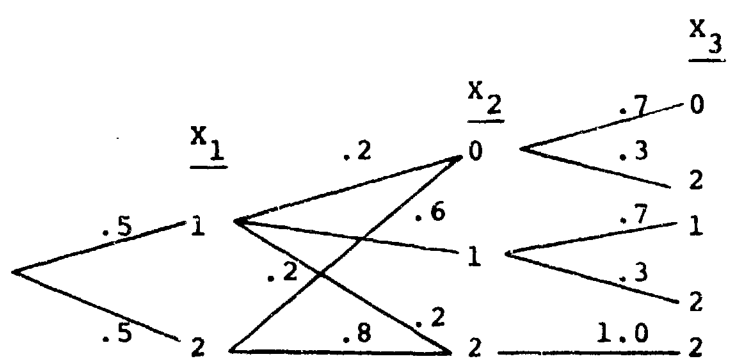


FIGURE 2.5. MODIFIED TREE.

A similar procedure can be applied to operator 3 - that is, we can simultaneously add signal 3 and drop signal 2 - and hence end up with just the required tree branch terminations which represent the distribution of signal 3 alone.

The branch combination technique is automatically applied by GO. The user indicates which signals are to be the final signals, and then the program decides at what point - if any - a particular signal can be dropped. We note that, for example, if we had directed GO to retain both signal 1 and signal 3 as final signals - that is, we wanted the joint distribution of both signals - the branch combination indicated in Figure 2.5 would not occur.

Using this technique allows us to handle problems with an arbitrarily large number of operators. However, if at some point the size of an intermediate distribution becomes too large (up to 3000 terms are permitted in most problems), the pruning technique must be used. Such storage overflow is normally caused by the presence of numerous active signals - that is, the number of signals in the joint distribution at some point is relatively large. This in turn occurs because of too many final signals or a very complex system structure which requires the retention of many signals for use as inputs to later operators.

The pruning technique makes use of a user-defined probability parameter called PMIN. As each new term of a distribution is formed, its probability is compared with PMIN, and the term is dropped unless the new term probability exceeds PMIN. In this manner the tree is always subject to pruning, and a judicious choice of PMIN will prevent an excessively large array size from occurring. If an array does become too large, GO will automatically increase PMIN temporarily in order to cut the particular distribution down to an allowable size.

Pruning obviously introduces some error into the analysis. The total pruning error is ultimately revealed by the difference between the sum of the probabilities in the final distribution and 1.0 (which would be the sums in the absence

of any pruning). It is, of course, impossible to proportion this total error among the terms of the final distribution - if we could do this, there would be no error. However, the total error certainly gives an absolute upper bound on the error in any particular term. Thus, we are certain that the tree probability of any term lies between the computed value and that value plus the total error.

A considerable part of the art of GO modeling lies in manipulating the model and PMIN in such a manner that the resulting total error is kept to an acceptably low level. In many cases several iterations are needed, but usually a satisfactory solution can be obtained.

It should be noted that because of branch combination, some branches of the complete (uncombined) tree whose probabilities would be less than PMIN may not be pruned. On the other hand, we are assured that any such branch whose probability is greater than PMIN will definitely not be pruned.

2.5 A Second Example

Our second example introduces the type 6 operator which can be used to model a normally open switch contact (other applications and interpretations are, of course, possible). The reader should consult the discussion of this type in Chapter 6. We also show some of the printout generated by the GO programs.

Presume that the subsystem in Figure 2.6 is an integral part of a larger system requiring two distinct electrical signals precisely sequenced in time. The GO chart of this subsystem using the standardized GO building blocks is shown in Figure 2.7.

In Figure 2.7 the various operators are identified by the sequential numbers assigned by the GO program as well as

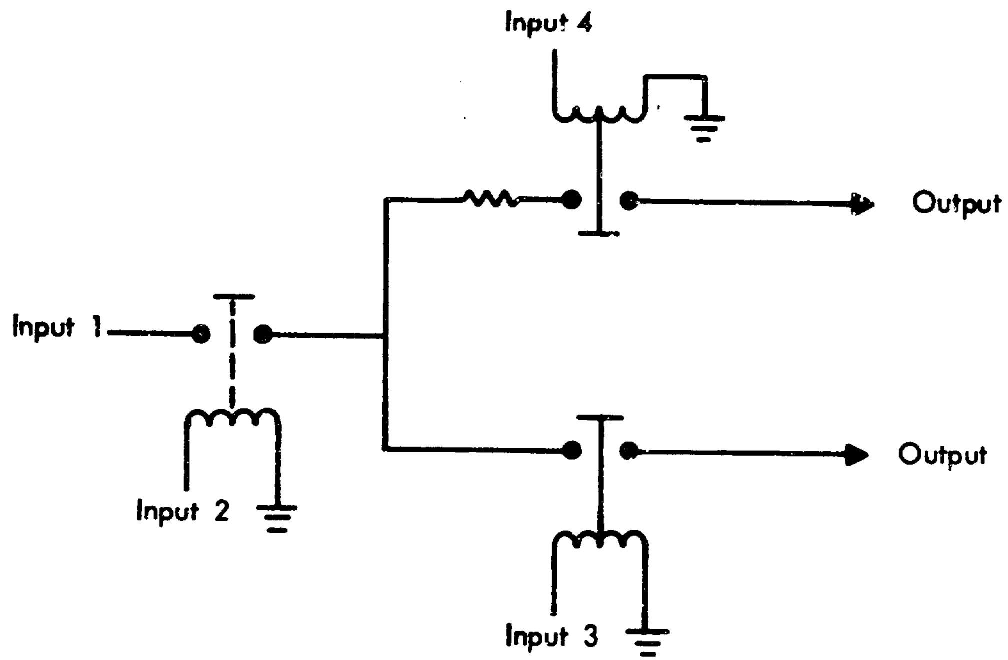
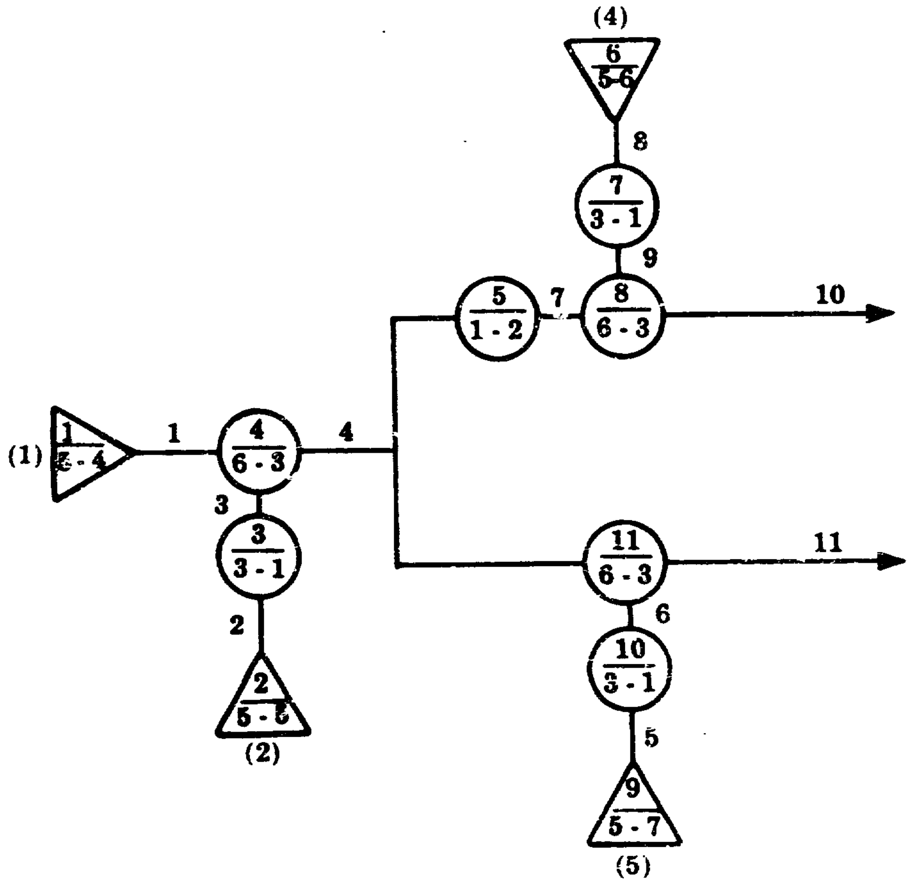
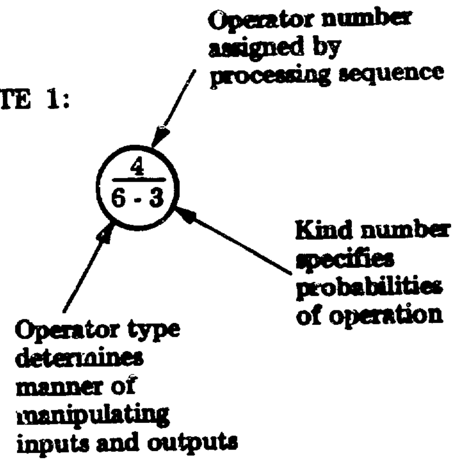


FIGURE 2.6. EXAMPLE SUBSYSTEM.



NOTE 1:



NOTE 2:

Triangular symbols annotated similar to circles with operator number always on top and type-kind numbers below regardless of triangle orientation.

NOTE 3:

Numbers in parenthesis outside triangles indicate most likely signal value (time). Numbers on line segments identify signals.

FIGURE 2.7. EXAMPLE SUBSYSTEM GO RELIABILITY DIAGRAM.

by the standardized GO type number and the specified kind number.

Data from the GO chart is punched on cards in the format shown in the "DATA" columns in Table 2.7. For example, the first entry of the table indicates that operator #1 is type 5, kind 4 and generates signal 1. The second entry defines operator 2 as type 5, kind 5 generating signal 2. Signal 3 is the output from the relay to actuate the normally open switch contact and is represented by a type 3 operator with kind 1 probabilities of operation and having signal 2 as input.

Table 2.8 lists the kind data specified by the user. Kind 1 is for a type 3 operator and requires three probabilities for the success, failure and premature operational modes of the relays modeled. Kind 2 is for a type 1 operator (the resistor of Figure 2.5) and requires probabilities for the success and failure of the resistor.

The kind data for the type 5 operators are somewhat more complex (kinds 4,5,6 and 7). For this problem, eight signal values (time points) are specified from 0 to 7 inclusive (∞=7). A time point of 0 identifies a premature occurrence and a time point of 7 the failure of a signal to ever arrive. The kind data for the type 5 operators must specify at which time points the generated signals arrive and the associated probabilities.

For this example, each type 5 operator was defined to generate a signal with high probability at one time point and fail to generate a signal with low probability (at time point 7). Hence the data for kind 4 specifies that there are two time points; that a signal is generated at time point 1 with probability 0.9 and at time point 7 with probability 0.1. (The sum of the probabilities must be unity). Similar kind definitions were made for the other type 5 operators.

TABLE 2.7. OPERATOR DATA.

OPERATOR DATA

OP	DATA				
1	5	4	1$		
2	5	5	2$		
3	3	1	2	3$	
4	6	3	1	3	4$
5	1	2	4	7$	
6	5	6	8$		
7	3	1	8	9$	
8	6	3	7	9	10$
9	5	7	5$		
10	3	1	5	6$	
11	6	3	4	6	11$
//12	0	10	11	$	

TABLE 2.8. KIND DATA.

RECORD	KIND DATA
1	1,3,0.8,0.1,0.1$
2	2,1,0.9,0.1$
3	3,6,0.8,0.1,0.1$
4	4,5,2,1,0.9,7,0.1$
5	5,5,2,2,0.9,7,0.1$
6	6,5,2,4,0.9,0.1$
7	7,5,2,5,0.9,7,0.1$

The computer output for this sample problem is shown in Table 2.9 and indicates that with the arbitrarily defined component probabilities of operation depicted in Table 2.8 the event of greatest probability of occurrence is "signal 10 occurs at time point 7 and signal 11 occurs at time point 7", $(10_7 11_7)$, with probability 0.373. All other operational states (joint events) and their probabilities of occurrence are likewise printed out. Because time point 7 represents failure to occur, the most likely event, $10_7 11_7$, represents failure of both signals 10 and 11 to arrive.

The entries in Table 2.10 are the appropriate sums of the entries of Table 2.9 and show the individual marginal probability distributions of the final signals. Thus the probability of the event "signal 11 arrives at time point 1" is 0.02916 without regard to what happens to signal 10, etc.

All GO models are generated and exercised as indicated in this example. Most such studies would require multiple runs with variations in the "kind" probabilities to determine system sensitivities and failure distributions. Supplemental runs may also be desired to produce input data for the Fault Finder programs (see the "Fault Finder Manual").

2.6 GO Modeling Philosophy

The GO procedure development has focused on economy, efficiency, and a quick response analytical capability. This has not been in disregard of mathematical rigor and systematic effort but with an enlightened acceptance of reality. That is, there is seldom, if ever, sufficient time, resources, capability and commitment to exhaustively model, accumulate the requisite component data for and finally analyze a system.

TABLE 2.9. FINAL DISTRIBUTION.

FINAL EVENT TABLE (INFINITY = 7)

	SIGNALS AND THEIR VALUES	
PROBABILITY	10	11
.0047239200	1	1
.0064035360	1	7
.0093195360	7	1
.0151165440	1	5
.0151165440	4	1
.0151165440	2	2
.0204913152	2	7
.0298225152	7	2
.0483729408	2	5
.0483729408	4	2
.0860635238	4	7
.1252545638	7	5
.2031663514	4	5
.3726592250	7	7

TOTAL PROBABILITY = 1.0000000000
TOTAL ERROR = .0000000000

TABLE 2.10. FINAL MARGINAL DISTRIBUTIONS

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	10	11
1	.0262440000	.0291600000
2	.0839808000	.0933120000
4	.3527193600	0.0000000000
5	0.0000000000	.3919104000
7	.5370558400	.4856176000

The efficiency of the GO modeling method results from five specific capabilities: (1) the ease of system characterization using standardized and generalized components; (2) the capability to include time and sequenced operations; (3) the simultaneous generation of the entire set of system operational modes; (4) the technique of combining events which differ only in no longer relevant details, and (5) the pruning of low probability events.

The use of standardized modeling elements provides for quick translation from schematic and configuration data to the computerized model. Note that in general the normal operational sequence is modeled which is convenient for system engineers to conceptualize. A major objective of this manual is to define and elucidate use of the standardized component types to model typical systems composed of common equipments.

If learning the symbology and rules seems burdensome, recall that these are the requirements imposed for employing any modeling procedure. The GO procedure can be employed expertly by anyone willing to learn the rules of its discipline.

The capability to account for timing sequences is a principal strength of the GO procedure. Whereas most other models assume only two-state components (i.e., good or dud status) the GO procedure incorporates three-state (premature, good, dud) and multi-state components as well as the capability to define and treat numerous time intervals and operational sequences. This is one of the most versatile features of GO. While equations and fault trees can be used to generate similar information, to do so is a tedious and error prone procedure at best because of the conditional nature of the events and requirements for repetitive computer runs or hand calculations with

properly altered (conditioned) inputs. The GO procedure produces such results in a single run and generates significantly more system information for the same amount of scientific labor than other procedures.

In creating a GO model in which signal values are related to time a fundamental consideration is the definition of the time points of interest. The use of the phraseology time point rather than time period is purposeful. The GO program does not simulate time discretly. It treats events which are defined to occur at specific instants in time, i.e., time points.

The selected time points hardly ever denote equally spaced time intervals. Instead they are usually defined (1) to specify sequential system response points or (2) to differentiate between various initiating causes which may occur simultaneously.

For example, assume that it is desired to determine if a system functions or fails to function at a given instant in time. In this case, only two time points need be specified, 0 and 1. The time point of intended operation is time 0. A signal value (time) of 1 would represent the failure of the signal to occur, or that the event represented by the signal never occurred.

If a given system has two external inputs whose normal occurrences are temporarily distinct, four time points of interest might be specified.

Time Point	Definition
0	System operates before receiving any inputs (premature).
1	Time of arrival of 1st external input.
2	Time of arrival of 2nd external input.
3	Never (failure of system to operate).

By examining the events in the output array, the times of system responses and their probabilities are readily identified and related to the arbitrarily defined time reference. Hence, in constructing the GO model an underlying chronology is assumed and required for correct interpretation of the results.

The signal values have been defined in a general way and could represent various quantities - dollars, flow characteristics, counters, etc. However, the original motivation was, and by far the most common application to date has been, to define time points of interest. Consequently, the essential equivalence of signal values and signal times in this treatise reflects this usage.

To illustrate this capability refer to the GO output in Table 2.9 from the example system of Figure 2.6 where 8 time points (0 through 7 inclusive were defined). Note that there are 14 resultant joint events showing the time combinations when output signals 10 and 11 are generated. The normal sequence (the designed operation) is to have a signal 10 output at time point 4 and a signal 11 output at time point 5. This event occurs with probability 0.203.

All other combinations are essentially anomolous. The most likely event is that neither output signal ever arrives; i.e., $10_7 11_7$. The least likely event is that both signals are premature at time point 1, event $10_1 11_1$. Various other combinations were registered with probabilities of occurrence falling between these extremes.

In many instances these anomolous joint events may be of little consequence. Conversely, however, output times are often extremely critical to system operation and must be analyzed with precision. The GO procedure permits this

refinement almost casually and produces the entire joint event time or value distribution as standard output.

In modeling systems with numerous elements, astronomical numbers of event combinations can easily occur. This imposes severe economic penalties in computer run times and storage requirements when attempts are made to retain all such combinations. In practice, some approximation scheme must, almost inevitably, be employed to provide an economical result.

In the GO procedure as presently implemented the quantitative results calculated are produced from the balanced interaction of three factors: (1) computer run time; (2) computer core storage available and (3) accuracy. As previously noted, these quantities are controlled in two ways: (1) by selectively pruning the event tree based on the magnitude of the probability of occurrence of the event combinations as they are generated, and (2) by combining like events when signals are deleted (branch combination).

The parameter PMIN specifies the threshold of retention. If the value for PMIN were specified as 1×10^{-10} all event combinations whose probabilities exceed that value would be retained. If there were not sufficient core storage to manipulate the remaining terms the value of PMIN would be automatically increased to eliminate additional terms and keep the problem tractable. This truncation procedure enhances efficiency but of course, reduces accuracy. Each run therefore requires a judicious selection of the PMIN value to balance accuracy versus cost.

For these reasons, employing the GO procedure cannot be an unconscious act. Each application will require consideration of these tradeoff effects to maximize the value of the calculations and minimize the costs. Familiarity with the GO procedure will provide the user with the skill and intuition to take advantage of this feature which makes the analysis of most systems tractable. Without these approximation procedures few problems of any complexity could be handled.

Use of the GO procedure essentially transfers the tedium and complexity of mathematical manipulation and calculation to the computer. Freed from this major burden, the analyst may concentrate on understanding system operation, insuring the adequacy of the model, gathering and validating data, performing numerous excursions, and interpreting results. The accuracy and thoroughness of such analyses are significantly increased over similar work of commensurate duration performed without this capability.

CHAPTER 3
COMPARISON WITH OTHER METHODS

3.0 INTRODUCTION

In this chapter several methods of system analysis are briefly described and their capabilities compared with those of GO. Although GO is a powerful tool for answering many questions about the operation of a system, it does not do everything. Consequently an awareness of its limitations and of possible alternatives can be of considerable benefit to an analyst.

3.1 Lumped Parameter Models

When only a single entity (a complex equipment of many constituent parts or a single element) is considered, one and two parameter statistical distributions are commonly used to model behavior and assess capabilities.

For example, assume that one purchased a new automobile, and from manufacturer data the salesman indicated that the mean time to failure requiring significant repair was three years. Presumably he meant that if one performed all normal maintenance, oil changes, lubrications, periodic check of the transmission oil, spark plug and point replacements, etc., that on the average it would be three years before major maintenance was required.

If one had some experience with, or had accumulated data from prior automobile models, he might have noted that the time-to failure distributions were approximately exponential in nature. That is, the probability that a given vehicle from the total number of like automobiles manufactured does not fail from the time of purchase to time t

could be expressed with one parameter in the form $P_{NF} = e^{-t/\theta}$ where θ is the mean time to failure (MTTF). The probability that it fails prior to time t is then $1-e^{-t/\theta}$.

With the information given one could calculate these probabilities, draw the cumulative distribution and make some judgements about the desirability and economics of such a vehicle. Suppose he did so and calculated the values as shown in Table 3.1.

TABLE 3.1. EXPONENTIAL FAILURE DISTRIBUTION.

Time t (Years)	Probability Vehicle has not Failed P_{NF}	Probability Vehicle has Failed $1-P_{NF}$
0	1.000	0.000
1	0.717	0.282
2	0.513	0.487
3	0.368	0.632
4	0.264	0.736
5	0.189	0.811
6	0.135	0.865
7	0.097	0.903
8	0.069	0.931
9	0.050	0.950
10	0.036	0.964

With this information one can assess the chances of his vehicle requiring major repair from wearout or malfunction (excluding accidents) for its anticipated serviceable life. There is, for example, a 51% chance that it will not require such repair in the first two years of operation. Since "the decisions of a wise man are determined by probabilities" such data, if correct, allow one to make rational decisions

now which affect and determine future circumstances. Of course, other distributions might have been equally applicable to this fictitious exercise. Some frequently used ones other than the exponential are the normal, lognormal, gamma, Weibull and uniform distributions each requiring the definition of two or three parameters.

The use of such distributions presumes the existence of valid parameter data, usually from a limited number of sample test results on the system or equipment from which the parameter values are inferred. On systems being designed or newly fabricated such data is not available. In these cases estimates for the parameters from similar operational equipments are often made by comparison and extrapolation. They are at best educated guesses, and if an incorrect distribution has been selected, the resulting estimates may be misleading. It should likewise be clear that if the effects of repair were included, a different and perhaps more realistic model would be required.

These distributions are often called "lumped parameter" models because one or two parameters account for the effects of all underlying failure causes, different environments, failure modes and numerous subsystem failures. Because of this integrating feature they do not provide visibility about the nature or causes of failure nor do they permit design alteration studies to enhance system capabilities. On large systems there are often significant data about constituent parts which are ignored or disregarded in the lumped parameter models.

An additional weakness is that the events of greatest interest - abnormal hazardous failures, malfunctions, extended operation - are generally quite unlikely to occur and estimates of the probabilities of occurrence of such events

are determined from the tails of these distributions. Thus, while events which normally occur or are quite likely to occur can be adequately modeled by a number of distributions, the use of improper distributions or parameters can give wholly fictitious inferences about events of small probability.

For these reasons additional procedures have been developed to correctly synthesize and incorporate data about individual components and subsystems into a representation or model of a larger mechanism. Such procedures then permit examination of the combined effects of estimates or assumed capabilities of the constituent elements, take advantage of additional information and allow enhancing design excursions.

These capabilities and sophistications are acquired only with complicating algorithms and the requirement to treat and account for astronomical numbers of combined events. A small system comprised of twenty components which are either functional or nonfunctional would generate 2^{20} (1,048,576) system states which must be assessed to completely characterize system operation.

Despite the proliferating complexities, the desirability and economy of such information obtained from simulation procedures as contrasted with the economic penalties of actual system testing of large complex systems have produced numerous variant modeling procedures providing these capabilities. These have become the stock in trade of the reliability and availability disciplines and practitioners whose techniques require increasing sophistication, capability and economy as the numbers, complexities and costs of systems escalate along with increasing awareness of, and sensitivities about, risks to society.

3.2 Equation Writing Techniques

Historically the equation writing technique has been employed extensively and is probably the best known and most widely used integrating concept. The probabilistic responses of the operational modes of each component or subelement are parameterized and incorporated into a polynomial equation in n variables representing the system event of interest (failure, success, premature, unsafe mode, etc.).

Often a reliability block diagram of the system is created. It is an abstraction, a model, which aids the analyst in writing the commensurate equations as a secondary model.

Consider the following parallel subsystem of a Network Feeder Bus (Figure 3.1) from a proposed Federal Aviation Administration Critical Power Distribution System. Two of the four identical feeder bus elements (assumed to be either good or dud) must provide proper power for successful subsystem operation. With a constant failure rate, λ , assumed for each element, the reliability for each element at time t is $r = e^{-\lambda t}$. (Here again the exponential distribution has been assumed as a valid model, but in this case for the individual elements rather than the composite system. In this case λ is the failure rate, the reciprocal of the MTTF).

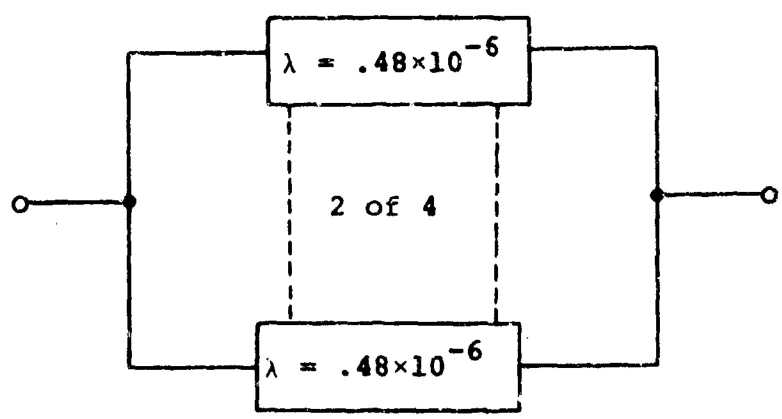


FIGURE 3.1. NETWORK FEEDER BUS.

Letting r_i be the event that the i^{th} element is good, the entire probability space is defined by the Boolean expression

$$\Omega = (r_1+\overline{r}_1)(r_2+\overline{r}_2)(r_3+\overline{r}_3)(r_4+\overline{r}_4),$$

where $\overline{r}_1$ is the failure of the i^{th} element.

When this expression is expanded, it becomes

$$\begin{aligned}\Omega = {} & r_1r_2r_3r_4 + r_1r_2r_3\overline{r}_4 + r_1r_2\overline{r}_3r_4 + r_1r_2\overline{r}_3\overline{r}_4 \\ & + r_1\overline{r}_2r_3r_4 + r_1\overline{r}_2r_3\overline{r}_4 + r_1\overline{r}_2\overline{r}_3r_4 + r_1\overline{r}_2\overline{r}_3\overline{r}_4 \\ & + \overline{r}_1r_2r_3r_4 + \overline{r}_1r_2r_3\overline{r}_4 + \overline{r}_1r_2\overline{r}_3r_4 + \overline{r}_1r_2\overline{r}_3\overline{r}_4 \\ & + \overline{r}_1\overline{r}_2r_3r_4 + \overline{r}_1\overline{r}_2r_3\overline{r}_4 + \overline{r}_1\overline{r}_2\overline{r}_3r_4 + \overline{r}_1\overline{r}_2\overline{r}_3\overline{r}_4\end{aligned}$$

Thus each of the 16 possible subsystem states is explicitly defined. Those having two or more reliable elements result in subsystem success, R_1, which is thus defined by

$$\begin{aligned}R_1 = {} & r_1r_2r_3r_4 + r_1r_2r_3\overline{r}_4 + r_1r_2\overline{r}_3r_4 + r_1r_2\overline{r}_3\overline{r}_4 \\ & + r_1\overline{r}_2r_3r_4 + r_1\overline{r}_2r_3\overline{r}_4 + r_1\overline{r}_2\overline{r}_3r_4 + \overline{r}_1r_2r_3r_4 \\ & + \overline{r}_1r_2r_3\overline{r}_4 + \overline{r}_1r_2\overline{r}_3r_4 + \overline{r}_1\overline{r}_2r_3r_4 .\end{aligned}$$

Because the summed terms are mutually exclusive and the factors within each term are assumed to be independent, we can obtain the probability of the event R_1 by replacing each r_i by r (the element success probability) and $\overline{r}_i$ by $1-r$ and applying the usual rules of numerical algebra. In addition we may replace r by its time function $e^{-\lambda t}$. We get

$$\begin{aligned}P(R_1) &= 3r^4 - 8r^3 + 6r^2 \\ &= 3e^{-4\lambda t} - 8e^{-3\lambda t} + 6e^{-2\lambda t} .\end{aligned}$$

Thus the polynomial equation for subsystem success can be written as a function of the common failure rate, λ , and the time elapsed. The fact that each element has an assumed exponential failure distribution and is time dependent is incidental to the general procedure. Time may have been excluded and other distributions or numerical estimates might be similarly used.

Once the equations are obtained, quantitative estimates can be readily calculated, sensitivity studies performed and various system design characteristics noted as a function of elemental component configurations and capabilities. Equations can be written to define failure events or premature events as well as success events. The complete generality of the procedure recommends it, and it is used in conjunction with many other procedures by way of documentation and rigorous expression.

Unfortunately the complexities of many systems and the quantities of elements to be modeled make the equation-writing technique difficult to apply. Inordinate amounts of analyst time are required to employ it, and it fares badly when contrasted with more recent computer-aided procedures.

3.3 Fault Tree Technique

The fault-tree technique, initially popularized by the Bell Telephone Company in the early 1960's, has been widely publicized by the boeing Company and others in the last decade. It has become a common procedure for performing safety and reliability studies.

"The fundamental concept in fault tree analysis is the decomposition of a physical system into a logic diagram, or fault tree, in which certain specified causes lead to one specified "TOP" event of interest. This logic diagram is

constructed using the symbols found in Table 3.2. The two basic units involved are AND and OR gates. Another, less often used element, is the NOT gate." [1].

It is important to recognize that a given tree defines only one system event. Different trees are required for each different event studied and their conditional nature (i.e., different trees use the same components) render it difficult to comprehensively analyze complex systems.

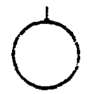
A typical fault tree is shown in Figure 3.2. It depicts the combinations of basic events (identified by number) which produce the top event. It also demonstrates the use of standard fault tree symbols which are defined in Table 3.2.

As in all procedures a crucial step in employing the fault tree technique is the creation of the logic diagram. The art, skill, knowledge and experience of the analyst are fundamental prerequisites to a good fault-tree model.

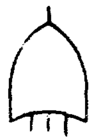
Once the fault tree has been created it is manipulated by any of a number of computer programs. Two-step evaluations use the PREP, SETS, ELRAFT, TREEL, MOCUS, MICSUP, ALLCUTS and other computer codes which find the minimal cut sets (sets of components which must all fail to induce the TOP event). Once found, these are evaluated numerically with the Idaho Nuclear codes KITT1, KITT2 [2] in standard use or similar ones. Direct evaluations are performed using a number of other programs which do not produce the cut sets explicitly to evaluate the tree. Among these are

NOTED - (UKAEA Authority Health and Safety Branch, 1971)

GO - (Kaman Sciences Corporation, 1968, 1976)

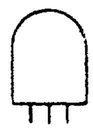


System component or basic fault event.



Inputs

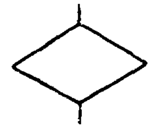
OR Gate.
This gate is in the failed state if at least one of its inputs is in the failed state.



Inputs

AND Gate.
This gate is in the failed state only if all of its inputs are simultaneously in their failed states.

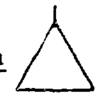
Event Descriptor.
The rectangle is used to describe the event represented by a gate.



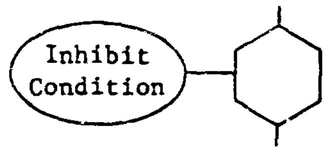
The diamond is used to represent a fault event which is not developed further due to lack of information.

Out

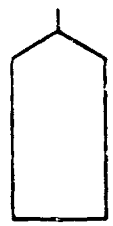
in



Transfer Symbols.
These symbols are used to transfer an entire part of the tree to other locations on the tree.



Inhibit Gate.
This represents an event which occurs with some fixed probability of occurrence. The inhibit gate is in the failed state only if its inputs are in the failed state and the inhibit condition has the inhibit condition has occurred.



The house represents an event which is normally expected to occur. It is treated as a switch on the tree, and is set on or off by the PREP user.

TABLE 3.2. FAULT TREE SYMBOLS (REFERENCE 2).

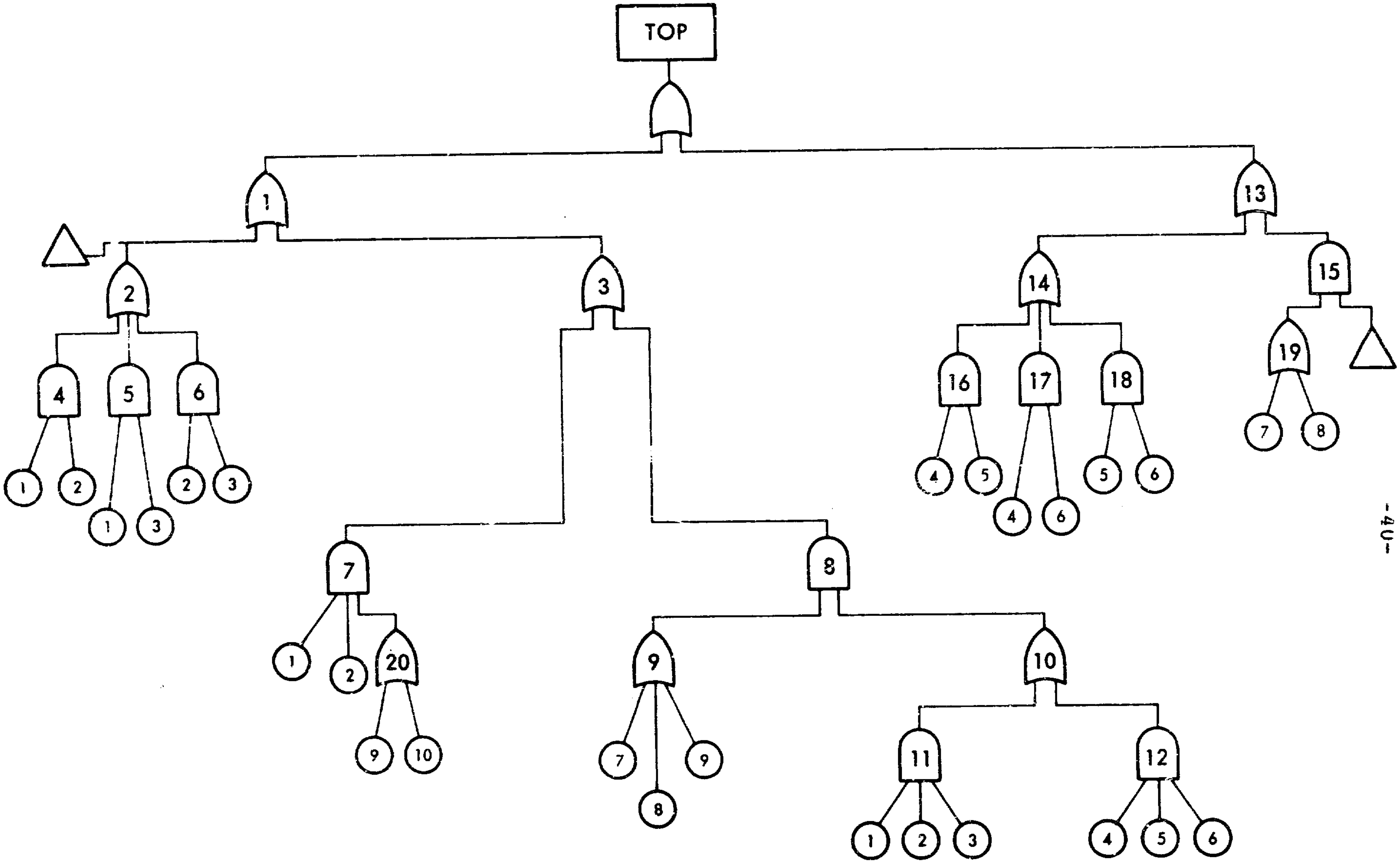


FIGURE 3.2. SAMPLE FAULT TREE (REFERENCE 2).

BAM - (Science Applications Incorporated, 1975)

ARMM - Automatic Reliability Mathematical Model - (North American Aviation, 1966)

SAFTE - (Nuclear Engineering and Design, 1970)

SAMPLE - (USAEC, 1974)

REDIS - (International Atomic Energy Agency, 1975) [1]

While fault trees have been widely employed, they are not without their critics. Fussell, Powers, Bennetts and Yellman point out "that fault trees are costly to develop, require skilled analysts, are not good at accounting for mutually exclusive events or common-mode failures, are prone to oversight and omission, often rely on poor assumptions" and require a lot of time.

"Claims to the contrary notwithstanding, construction of a fault tree is not a very systematic process. It is really quite subjective, and different analysts come up with different fault trees for the same system and mission. Furthermore, the format of the fault tree is such as to overwhelm with information, rather than to break a problem down into manageable parcels. These characteristics of the fault tree create difficulties for both the analyst and those with whom he must communicate.

"Of even more importance, there is a large class of systems in which basic (component-level) events cannot be expressed (sequence) s-independently. With the exception of repeated identical events, fault trees do not satisfactorily handle s-dependencies. The importance of s-dependencies is noted in the more perceptive literature and may be verified by people who pay attention to how systems really fail. Unfortunately, reliability analysts

too often assume s-dependency away, encouraged perhaps by such statements as 'when component reliabilities are high, this assumption enables approximate results to be obtained for s-dependent units'. This is misleading and usually untrue. Another ploy is to define away s-dependency by the euphemism 'primary element'. This is meant to implicitly excluding dependent events without the embarrassing necessity of actually saying so. Unfortunately, real life is seldom so accommodating. Typical system characteristics which give rise to s-dependencies include passive redundancy, active redundancy accomplished by sensors and switching devices, mutually-exclusive malfunctions, secondary malfunctions, stress levels changing following malfunctions, multiple malfunction, resulting from common external stresses, component malfunctions related to human errors, and human errors related to either previous component malfunctions or other human errors." [3]

Yellman proceeds to discuss his "Event Sequence Analysis" scheme and concludes by "anticipating somewhat less than hearty acceptance from some who teach the art of constructing fault trees, and others who have a fetish for using computers whether they need them or not."

The criticism of fault tree construction being quite subjective is not unique to that procedure. Most models include subjective elements which are difficult to eliminate, define or justify.

The criticism that the fault tree format overwhelms with information and creates communication difficulties must be tempered by considering the application and the user. The information is often necessary and helpful, and the fault tree presentation may excel other schemes in its presentation for specific analyses.

To account for sequence dependent events usually requires or would require more resources to obtain sufficient information than can be committed to perform the assessment. Detailed failure mode and effect analyses (FMEA), shorting analyses, fault propagation studies, actual abnormal environment testing, etc., would be required to rigorously treat sequence dependent events and these studies are inordinately expensive. Hence, the criticism is valid but may be academic.

The fault tree procedure and most other reliability procedures assume the independence of event occurrence or component malfunction. Some dependencies are recognized and treated, but this is difficult at best in a fault tree model.

Those unfamiliar with computer intricacies and capabilities or those who have had negative exposure might impute a fetish to those who employ computers routinely. It is true that many of the existing fault tree computer programs are inefficient and costly to exercise. While the criticism may be valid in specific instances, it cannot be justified in general; and to persist in the assertion categorically is to negate a vital capability which has made the analysis of complicated systems feasible.

One significant and very real problem for all modeling procedures is that the numbers of events or sets of interest become inordinately large for even relatively small systems. This fact places severe restrictions upon the analysis - forcing the use of approximation techniques or intolerable computer run times. One severe criticism of some fault tree computerized evaluation schemes is their inefficiency. Much work has been done and continuing effort is being expended to enhance the capabilities and render the processing algorithms more efficient and economically feasible.

At the present time, knowledgeable observers estimate that a large proportion of all safety and reliability modeling and assessment work is being performed with some variation of the fault tree procedure. This is no doubt due to the wide exposure which it has received and its basic simplicity. The relative obscurity to date of other competitive methods and their procedural requirements may account for the widespread utilization of the fault tree technique despite the insufficiencies noted.

3.4 Tests And Use Data

Of course, the best procedure for determining reliability is actual experience data with the equipment. Data kept for continuously operating equipments identifies critical parts, wearout rates, mean times to failure and repair, probabilities of operating for specified periods, mean down times, overall availability, etc.

For expendable or one-shot systems the proportion of successes to the total used provides an estimate of the actual capability of the identical elements of a homogeneous population of equipments. The larger the number tested, the more certain are the estimates.

Unfortunately, this procedure for determining the reliability of a complex system is usually not an option because decisions to manufacture a system often depend on the expected reliability it will have. Generally speaking, the costs of such equipments are so high and the overall cost effectiveness so dependent upon the ultimate reliability or availability that these parameters must be determined accurately during the preliminary design phase. Consequently, a "use only" procedure for determining system reliabilities is viable only after the systems have been employed or expended.

3.5 Simulation

Another common procedure is a simulation of the system being studied. It is often employed when specific equations or other models cannot be generated easily. Typically some assumptions about the distributions of random occurrences are made, and sequences of random numbers are drawn to depict the failures, repairs and other events of interest.

"'Monte Carlo' is the code name given by Von Neumann and S.M. Ulam to the mathematical technique of solving problems too expensive for experimental solution and too complicated for analytical treatment. Originally, the concept referred to a situation in which a difficult but determinate problem is solved by resort to a chance process. A simple illustration of the idea is the problem of finding the surface area of an irregularly shaped lake enclosed in a rectangle R, as shown in Figure 3.3. Instead of computing the area by geometrical methods, we apply the following chance process. A large number of pebbles are projected by means of a catapult whose elastic is stretched to randomly varying lengths at angles which too vary equally randomly.

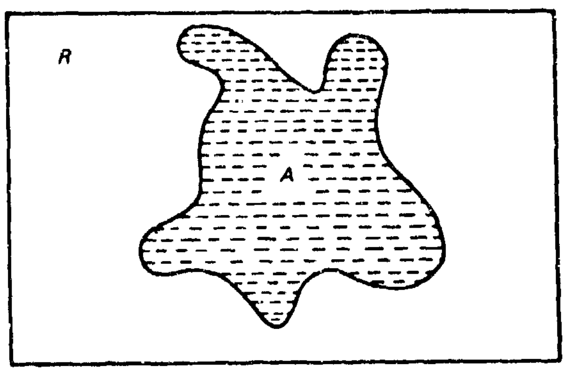


FIGURE 3.3. LAKE OF AREA A ENCLOSED IN RECTANGLE OF AREA R. FROM REFERENCE [10].

The area A of the lake then is simply

$$A = R\left(\frac{\text{Number of pebbles falling in water}}{\text{Total number of pebbles projected}}\right).$$

The underlying rationale of the process is that the probability that a pebble will fall in water, that is, the frequency ratio of throws falling in the water to the total number of throws, is equal to the ratio of the areas of the lake and its enclosing rectangle. We have here a probabilistic process that yields us the solution of a determinate problem, namely, the area of the irregular lake." [4]

"The ultimate test of any real system is how well it actually operates, where the words, 'how well' may include various criteria such as cost, availability, reliability, etc. Most system designs are subjected to a variety of analyses which throw some light on these matters, but it becomes extremely difficult to put all of these together in a meaningful manner. For example, one can make extensive reliability and maintainability analyses of the individual components of a system using standard methods, but it is not always clear just how these can be incorporated into a system model which might - and probably should - also include information concerning administrative, logistic, and operational details. This is not to say that such a comprehensive system model is impossible, but the difficulties can become very severe, and frequently one is forced to impose theoretical constraints which, while permitting an answer, often solve an irrelevant problem.

"A Monte Carlo simulation of the operation of a system allows one to incorporate almost any feature and does not require the use of particular random variable distributions in order to obtain analytical convenience - e.g., exponential distributions are almost universally used. The

price extracted for this generality is that the resultant answers - e.g., estimates of system availability - are uncertain. That is, they are statistical estimates of certain system parameters rather than the parameter values themselves. By judicious use of statistical methods, however, one can obtain confidence bounds in most cases. In short, a Monte Carlo simulation will produce uncertain answers to the correct problem rather than exact answers to the incorrect one.

"Fortunately, the variability introduced by a Monte Carlo procedure can be made arbitrarily small by simply running the simulation a sufficiently long period of time. The accuracy of a Monte Carlo simulation improves as the square root of the number of trials. Thus, the deficiency previously mentioned can be overcome assuming that the cost of sufficient simulations is not prohibitive.

"One of the principal advantages of the Monte Carlo approach is its basic simplicity. This is offset to some extent by the necessity for carefully interpreting the results. Nevertheless, for many situations, it provides a powerful tool for understanding the operational details of a system and for investigating the effects of various changes to the system." [5]

At least two general reliability-availability simulation computer programs are available, the General Dynamics Operational Analysis Model (OAM)[6] and the Kaman Sciences System Operation Simulation Model (SOS)[7]. Results from the SOS program characterize a number of system parameters including component MTTF, component MTTR, system MTTF, system MTTR, the system failure and repair distributions, system availability, critical components and simultaneous repairs.

One criticism of such models is that failure and repair distributions for each component must be known to create them. In general, such data is not available, so the distributions are assumed based on extrapolation on data from similar components. These uncertainties reduce the credibility of and confidence in the results of such simulations, but the criticism is equally applicable to virtually all assessment procedures.

3.6 GO And Some Comparisons

The GO procedure was developed as an automated extension of equation-writing and modified Boolean algebra manipulations by researchers at Kaman Sciences Corporation in 1968. It is a classical reliability procedure where piece part and component data are synthesized to characterize the probabilistic behavior of a composite system.

The central determination of any analysis is its objective, the answer to the question, "What information is required about a system?" The corollary question is "How can it be economically obtained or inferred?" Table 3.3, Comparison of Reliability Procedures, provides a general description of the required inputs and resultant outputs generated by the specified procedures.*

Since all these procedures except the lumped parameter models require an accurate knowledge of the system to be modeled - the constituent elements, their operating environments, their operational and failure modes and probabilities

* The capabilities of the Fault Finder programs have been included in the GO category. The Fault finder requires the use of GO but not conversely. See the "Fault Finder Manual" for complete details.

of occurrence, their mutual interaction and composite configuration - , the comparisons in Table 3.3 are made assuming that such information is equally required and available for all procedures. The comparisons become meaningful in the nature, extent and economy of generating valid estimates of desired parameters.

As summarized in the table, the GO procedure is applicable to:

1. Determine the probability that a system has not failed to time t where the constituent elements have known time dependent failure distributions and no repairs are effected. Both point estimates (a single run) and the entire system reliability function (multiple runs) can be generated. That function, once determined, can be integrated to find the system MTTF.
2. Determine the availability of a system composed of independent elements whose availabilities are known. Component availabilities are derived as a function of MTTF and MTTR estimates which are assumed independent.
3. (Fault Finder) determine the minimal cut sets of up to four elements for selected events. This capability is more general than a fault tree since a single GO run creates n output joint-events any one of which or any combination of the several can be selected to obtain the fault sets. There could thus be a 2^n fault events treated and the cut sets determined.
4. Calculate point estimate reliabilities of one-shot systems (essentially the same as #1 above, no

TABLE 3.3. COMPARISON OF RELIABILITY PROCEDURES

MODEL OUTPUTS AND CAPABILITIES	Lumped Parameter	Equation writing	Fault Tree	Use Only	Monte Carlo	GO
1. Probability of operating successfully for specified time period						
a. Point Estimate	X	X	X	X	X	X
b. Entire Distribution	X	X	X	X	X	X
2. Operational Modes						
a. Simultaneously Identified				X		X
b. Likelihoods of Occurrence Simultaneously Determined				X		X
3. Time Dependencies Modeled		X (?)	X (?)	X		X
4. Physical Parameters Denoted				X		
5. Sequence Dependent Events Incorporated			X (?)	X	X	X (?)
6. Minimal Cut Sets Determined			X			X
7. Critical Components Identified		X	X	X	X	X
8. System Characteristics						
a. MTTF						
1. Failure Distribution without Repair			X(?)	X	X	X
2. Point Estimate without Repair			X(?)	X	X	X
3. Failure Distribution with Repair				X	X	
4. Point Estimate with Repair				X	X	
b. MTTR						
1. Repair Distribution				X	X	
2. Point Estimate				X	X	
c. Availability						
1. Point Estimate			X	X	X	X
2. Confidence Interval			X (?)	X	X	X(?)
9. Spare Parts Required				X	X	

(?) Indicates limited or special treatment capability.

TABLE 3.3. COMPARISON OF RELIABILITY PROCEDURES (Continued).

	COMPONENT INPUTS REQUIRED*					
MODEL OUTPUTS AND CAPABILITIES	Lumped Parameter	Equation Writing	Fault Tree	Use Only	Monte Carlo	GO
1. Probability of operating successfully for specified time period						
a. Point Estimate		Point Estimates	Point Estimates	Actual Data	Failure Dist.	Point Estimates
b. Entire Distribution		Failure Dist.	Failure Dist.	Actual Data	Failure Dist.	Failure Dist.
2. Operational Modes						
a. Simultaneously Identified				Actual Data		Point Estimates
b. Likelihoods of Occurrence Simultaneously Determined				Actual Data		Point Estimates
3. Time Dependencies Modeled						
4. Physical Parameters Denoted		*	*	Actual Data	*	*
5. Sequence Dependent Events Incorporated					Sequence Depend.	
6. Minimal Cut Sets Determined			System Config.			Point Estimates System Config.
7. Critical Components Identified		Point Estimates Partial Deriv.	Cut Sets	Actual Data	†	Point Estimates System Config.
8. System Characteristics						
a. MTTF						
1. Failure Distribution without repair			Failure Dist.	Actual Data	Failure Dist.	Failure Dist.
2. Point estimate without repair			Failure Dist.	Actual Data	Failure Dist.	Failure Dist.
3. Failure distribution with repair				Actual Data	Failure Dist. Repair Dist.	
4. Point Estimate with repair				Actual Data	Failure Dist Repair Dist	
b. MTTR						
1. Repair Distribution				Actual Data	Failure Dist. Repair Dist.	
2. Point Estimate				Actual Data	Failure Dist. Repair Dist.	
c. Availability						
1. Point Estimate			Availabilities	Actual Data	†	Availabilities
2. Confidence Interval				Actual Data	†	
9. Spare Parts Required				Actual Data	† Oper. Period	

* System configuration & operating characteristics

† Failure Distribution
Repair Distribution
Slack Times
Start up Time
Shut Down Time
System Configuration

repair) using point estimate probabilities rather than time dependent failure distributions. Incremental and rate of change sensitivities can be generated for each element.

As noted above, GO is generally not applicable to model the effects of repairs, slack times (i.e., time from component failure to system effect) start up and shut down times, etc. When these operational characteristics require definition in conjunction with system configuration and hardware capabilities a simulation procedure like SOS or one tailored to the system would correctly incorporate these features.

3.7 Summary

These brief descriptions of well-known reliability and availability procedures provide a reference framework from which to discuss the capabilities and utilization of GO. The GO procedure is subject to many of the same criticisms which have been made of those mentioned. The lack of adequate data cannot be remedied by performing a GO analysis. The statistical independence of component operations is typically assumed in a GO model although a qualified capability exists to account for and model dependencies.

Thus, while the unique capabilities of the GO procedure satisfy many of the objections and criticisms of other methods it cannot magically obviate deficiencies in knowledge of the system being modeled or computer computational capabilities. It is a procedure which optimizes the use of present day computers in synthesizing component information to address the operational characteristics of composite systems.

CHAPTER 4
THE GO PROGRAMS

4.0 INTRODUCTION

This chapter presents an overview of the three GO programs and the several data files. The chapters following this one will present the details of modeling techniques and data preparation.

The interested reader will find the details of the specific program algorithms in the "GO Program Manual".

4.1 General Description

The GO system contains three main programs: GO1, GO2, and GO3. A system flow chart showing the interrelationships of the three programs is given in Figure 4.1.

After a given problem has been analyzed and an appropriate GO chart constructed, the operator deck is prepared and processed by GO1. The operator data contains the type and kind and the identification numbers of all input and output signals of each operator. GO1 analyzes the logical structure of the problem, prints some information for the user, and creates the operator file. This file contains all of the problem structure data which is required by GO2 and GO3.

The kind deck contains the parameter data (probabilities and values primarily) for the various operator kinds. This deck is processed by GO2 (which also requires the operator file previously created by GO1). GO2 prints summary information and creates the kind file which will be used by GO3.

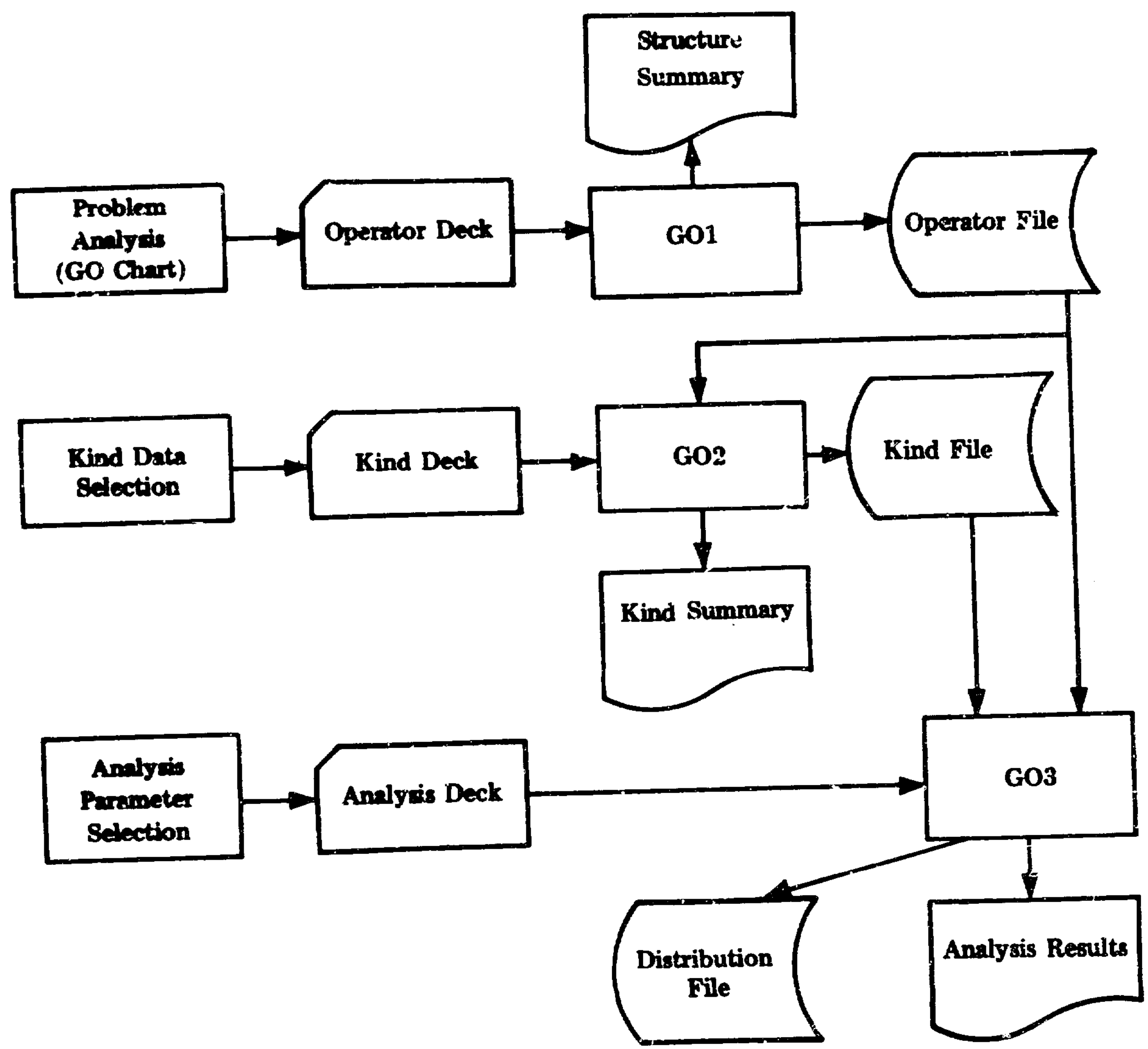


FIGURE 4.1. GO SYSTEM FLOWCHART.

Finally GO3 operates upon the data supplied in the operator and kind files and prints the results of the problem analysis. The <u>analysis deck</u> may consist of just two cards containing name and parameter data for GO3, but it may also contain additional kind data if sensitivity runs are to be made. (GO3 will optionally create the <u>distribution file</u> which serves as a data source for the Fault Finder Programs, whose use is covered in a separate EPRI report).

All three programs make extensive checks of the input data. These checks will catch most keypunching mistakes and many logical errors, but the user must be constantly on guard to avoid errors which can escape detection by the programs. (See Chapter 9 for a listing of the various error messages produced by the programs).

4.2 Data Files

The operator file is referenced as TAPE1 in GO1, GO2 and GO3. The kind file is referenced as TAPE2 in GO2 and GO3. Either or both of these files may be saved by appropriate control cards in order to avoid unnecessary repeat runs of GO1 and/or GO2.

A new kind file may be created without recreating a new operator file.

Both files contain creation dates and times which are printed by the using programs. This will permit the user to ensure that the proper files have been selected in case he has several saved files. In addition, GO3 will not permit a run to be made if the kind file was created from an operator file different from the one being used by GO3.

The distribution file is referenced as TAPE6 in GO3.

In general, these data files are of little direct interest to the user. The sophisticated user may, however, wish to do some external manipulation of the files. The exact contents of the file records are given in the "GO Program Manual".

4.3 Program Execution

For a given problem the programs must initially be executed in the sequence GO1, GO2, and GO3. Additional runs will not require executing GO1 if no operator deck changes are necessary and the operator file from an earlier run was saved. Similarly, GO2 need not be re-executed if no kind deck changes are needed and the kind file was saved. Any change in the operator file must be followed by a run of GO2 because the contents of the kind file are dependent upon those of the operator file.

Minor changes in kind data can be effected by making a sensitivity run of GO3 rather than recreating the kind file by GO2.

The control cards needed for a GO run depend upon the manner in which the programs are stored and upon local computer installation protocols.

CHAPTER 5
GO DATA

5.0 INTRODUCTION

This chapter deals with the specific details of the data required for and the printout produced by the GO programs. The reader is urged to make frequent reference to Appendix B which shows both the input and output data of a GO model analysis.

5.1 The GO Chart

The GO chart of a problem has the misfortune of being at the same time both unnecessary and of the utmost importance.

It is unnecessary in the sense that the GO programs, which actually produce all of the computational results, never "see" the GO chart directly. In fact, a reasonably competent analyst can usually prepare the GO data cards for a small, simple system without taking the time to draw a GO chart (although he probably will end up drawing one anyway if he has to pass the results on to other people).

On the other hand, the GO chart is of the utmost importance in the sense that it is in its construction that the full talents of the analyst are brought into play. The GO chart represents the analyst's view of the system, and it is here that the decisions concerning what to include, what to leave out, how to represent components and subsystems, and all of the other major and minor aspects of a successful model are made. For a large and complex system the GO chart creation is a task demanding the highest level of ability and requires a detailed and exhaustive knowledge of both the real system and the GO procedure. Once a well-constructed GO

chart is completed, the preparation of the data for the programs is largely a mechanical chore.

There are no inviolable rules for GO chart construction. The chart should bear a reasonably close resemblance to the original system - certainly in function and preferably in structure and layout - and should contain most of the information needed to prepare the program data. We will list a few suggestions which represent several years of experience and the analysis of many systems.

1. For ease in relating the model to the real system, the layout of the GO chart should conform as closely as possible with the system schematic or block diagrams that are used. In many cases an almost direct overlay is possible although the GO chart will usually contain at least a few logical operators which are not explicitly included on the system drawings.
2. Regular operators are represented by circles or, for type 5 operators, a triangle. Super-operators are represented by rectangles.
3. The type and kind numbers for each operator are written inside the operator symbol, separated by a dash.
4. Operator sequence numbers, which are ultimately assigned by GO1, may be written within the operator symbols after the GO1 run is successfully made.
5. Multiple inputs to operators in which different inputs have different functions should be distinguished in some manner. Some possible ways include labeling the inputs with letters or numbers

and, for two-input operators, using a full arrowhead on the "main" input and a half arrowhead on the "minor" one. Operator types 6, 7, 9, 13, 14, 16, and 17 should be treated in this manner but types 2, 10, and 11 can be left unmarked because all inputs to these are treated alike.

6. Arrowheads should be used to indicate the "direction of flow".
7. Signal identification numbers may be chosen arbitrarily within the range 1 to 9999, but be careful if supertypes are used because GO1 will generate the identification numbers for the internal signals on its own, and duplication of numbers will produce a fatal error in GO1. That is, the signal identification numbers must be unique.
8. Kind numbers may be chosen arbitrarily within the range 1 to 999.
9. Like any artistic creation, the final GO chart will probably be the result of several tries, so don't expect a perfect job the first time around.
10. Be careful to model the system as it is, and not how you would like it to be or think it should be.
11. Insofar as possible, keep the model simple. Such a model will frequently be adequate, and it is usually easier to expand a simple model than to contract a complicated one. Introducing unnecessary complexity into a model is a very common failing and should be resisted.
12. The use of supertypes can be extremely valuable in reducing analyst time and producing a much more

readable GO chart. Some caution is required however, because supertype usage can easily introduce excessive complexity and result in an overwhelmingly large model.

5.2 Card Formats

Three different formats are used for the cards making up the operator, kind, and analysis decks. These formats have been chosen for ease of use and are relatively trouble-free. However, some care is necessary and the rules given below must be scrupulously followed. Failure to do so will lead to detected errors at best and unpredictable results at worst.

5.2.1 Name Card

The first card of each deck (and sensitivity run subdeck) is a name card which contains any descriptive name the user desires. The card may be blank but must be present. The contents of the card are reproduced on the program printout, and the operator and kind deck names are written on the operator and kind files for identification purposes. With the exceptions noted in the following paragraph, all 80 columns may be used for the name.

The name cards for GO2 and the sensitivity run subdecks use the character in column 80 as a control. In GO2, if this character is nonblank, the perfect-case option is selected; otherwise, not. In a sensitivity run subdeck, a nonblank character in column 80 signals GO3 that parameter changes are to be made, and consequently GO3 will assume that a parameter card follows immediately; otherwise, a parameter card will not be expected.

5.2.2 Parameter Card

A parameter card is mandatory in the operator and analysis decks and is optional in sensitivity run subdecks as noted above. There is no parameter card in the kind deck. The parameter card is always placed just after the name card.

The purpose of the parameter card is to allow the user to change the default values of certain parameters within a program.

The parameter card uses the Fortran NAMELIST format and, with certain restrictions, is free-field.

Columns 1 through 8 <u>must</u> contain "^$PARAM^" (where "^ " is a blank).

A parameter change is effected by a <u>definition</u> which consists of a variable (parameter) name followed by an equal sign followed by the new parameter value followed by a comma. The four elements of a definition may be separated by blanks if desired, but imbedded blanks within an element are not allowed. Thus the definition "XX=10," may be written "XX^=^10^^," but not "X^X=1^0,".

The definitions are placed freefield on the card after column 8, and the list is terminated by "$" which <u>must</u> be present. The comma in the last definition may be omitted if desired.

Example (the first in column 1):

"^$PARAM^^^XX=1,^^YY=17.3,^^Z=0^^^$"

The numerical values may be written without regard for the Fortran "type". Thus "X=1" and "X=1." are equivalent regardless of whether X is an integer or a real variable.

Exponential notation may be used. Thus "X=0.00001" may be written "X=1.E-5" or "X=1E-5".

Although the need for it is unlikely, a parameter "card" may actually consist of two or more cards. The rules to do this are: (1) the first column of every card must be blank, (2) the "$PARAM" appears on only the first card, and (3) each definition must be contained wholly within one card. Thus our example above could be written as " $PARAM XX=1,^^YY=17.3," on the first card and "^Z=0 ^^^ $" on the second.

A parameter card may contain no definitions. Thus "^ $PARAM ^$" is a legal parameter card (it implies that the default values will be used for all parameters).

If several definitions are included, their order is immaterial.

Specific parameters to be used are discussed in later sections. The parameter name spelling must be exactly as shown; results are unpredictable otherwise.

5.2.3 Data Card

The operator and kind data are written in a free-format manner designed for this purpose. The data required for each type are specified in Chapters 6 and 7 and punched according to the following rules:

1. Data items (numbers) are separated from each other by one or more blanks and/or commas.
2. The items may be placed anywhere on the card.
3. The last item is followed by a "$" with intervening blanks and/or commas if desired.
4. The columns (if any) to the right of the terminator ($) may include a descriptive comment if desired. The entire contents of the card are

printed and a comment is frequently useful for documentation.

5. The "card" may actually be several cards. Only the last card will contain a terminator.
6. Data items must not have imbedded blanks.
7. Operator data must <u>not</u> contain a decimal point (all such data are integers).
8. Kind data may be written without a terminating decimal point if desired.
9. Each kind data item can contain no more than 10 characters.

5.2.4 EOR Card

An EOR (end-of-record) card contains the multiple punch 7/8/9 in column 1. Such a card is used to (a) terminate a supertype definition, (b) terminate a sensitivity run subdeck, and (c) terminate the operator, kind, and analysis decks.

5.3 <u>GO1</u>

5.3.1 Operator Deck

The operator deck contains the input data for GO1. Almost all of the data can be directly obtained from a complete GO chart. The deck contains (in this order):

1. Name card
2. Parameter card
3. Supertype subdecks (if any)
4. Operator records
5. Final signal card
6. EOR Card

5.3.1.1 Name Card

This card is described in Section 5.2.1. The name (up to 80 characters) will be reproduced on the GO1 printout and

TABLE 5.1. GO1 PARAMETERS.

Parameter name	Function	Possible values	Default
VALUES	number of signal values (1)	2 to 128	200 (2)
OPS	print operator data	0 means no 1 means yes	1 (yes)
SIGNALS	print signal table	0 means no 1 means yes	1 (yes)
ERRORS	number of errors before aborting the run	any positive integer	25
BIAS	automatic initial bias for super operators	0 to 9999	5000
INFIN	Value of infinity (3)	1 to 127	-

(1) "infinity" will be equal to this number minus one.

(2) This default value will produce an error but will permit the initial processing of the operator deck in order to check for other errors.

(3) This parameter may be used in place of VALUES; if it is specified, VALUES will be set at INFIN + 1.

will be written on the operator file and subsequently printed when the file is used by GO2 and GO3.

5.3.1.2 Parameter Card

The card format is discussed in Section 5.2.2. Each parameter name, function, possible value, and default value are given in Table 5.1. The default values are automatically assumed unless the user makes an explicit change on the parameter card.

It should be noted that a successful run of GO1 requires a user-supplied value of VALUES or INFIN. As a genral rule the other parameters should be left at their default values for at least the initial GO1 run of a problem.

5.3.1.3 Supertype Subdecks

Each supertype is defined by a <u>supertype subdeck.</u> The subdeck contains (in this order)

1. supertype declaration card
2. operator records
3. EOR card

The contents of the declaration card are detailed in Chapter 7. The operator records are the same as those given in the next section.

Actually a supertype subdeck may be placed anywhere within the operator deck as long as it occurs before it is called as a super-operator. A super-operator which is one of the operators within a supertype definition (a nested super-operator) is not actually called until the supertype itself is. However, it is generally good practice to place all supertype subdecks at the beginning of the operator deck (just after the parameter card).

5.3.1.4 Operator Records

The card format is discussed in Section 5.2.3.

A record for an operator (regular or super) will usually consist of a single card. See Chapters 6 and 7 for specific record contents.

5.3.1.5 Final Signal Card

This card uses the same free-format as the operator cards. The first data item is a zero (which serves as a flag), and this is followed by the identification numbers of those signals which are to be included in the final distribution of GO3. The order in which they will be printed in the final distribution is the same as the order in which they occur on the final signal card. Any unused signals - that is signals which are not used as inputs to an operator - will automatically be included in the final signal distribution even though they are not mentioned on the final signal card.

The values of the identification numbers of signals generated internally by super-operators are usually difficult to ascertain in advance, particularly if nested super- operators are involved. If such signals are to be included on the final signal card, making use of a user-assigned bias in the super operator record rather than the automatic bias may solve the problem, but frequently it is worthwhile to make a preliminary GO1 run in order to deterimine the desired signal numbers.

5.3.2 Output

The output from GO1 consists of the operator file and some informative printout.

The operator file is created for use by GO2 and GO3 and contains data relating to the structure of the problem.

The printout may be user-controlled to some extent by the parameters OPS and SIGNALS. Regardless of their values, the printout will include:

a. The contents of the name card (which will also be used to identify the operator file).

b. The date and time of the run.

c. A list of the values of the parameters VALUES, BIAS, OPS, SIGNALS, and ERRORS.

d. A summary of the problem structure which includes

   1. The number of operators (including those generated by supertypes).
   2. The number of signals.
   3. The maximum number of simultaneously active signals.
   4. The maximum size of the signal list (usually equal to the previous item).
   5. The number of signal values (=VALUES).
   6. The number of signal value bytes per computer word (in GO3).
   7. The number of computer words per term (in GO3).

e. A list of the final signal numbers which includes the final signal card list augmented by any unused signals.

If OPS = 1 (the default value), the contents of all of the operator deck cards following the parameter card will be listed along with the records of operators generated by supertypes. An identifier will be placed at the beginning of

each record and will be:

a. The operator number assigned by G01 for all regular operators and those generated by supertypes. These operators are the ones which make up the problem as it is actually sent to G02 and G03. The operator number used in certain optional printout from G03 and in the Fault Finder are the ones listed here. A supertype-generated operator record is also preceded by the nesting level of the operator; this is printed as (L=n) where n is the level.

b. "XXXX" for all supertype definition records, including the initial declaration. The definition will be determined by the message "----- ^^^ END OF SUPER TYPE n" where n is the supertype identification number on the declaration card.

c. "$$$$" for a super-operator record. This record will be followed by the records of the operators generated by the supertype.

d. "////" for the final signal card.

If SIGNALS=1 (the default value), the signal table will be printed. This table lists all signal numbers in increasing order and provides the following information about each signal:

a. The number, type, and kind of the source operator of the signal, i.e., the operator for which the signal is an output.

b. The number of each operator which uses the signal as an input. The last such number will be preceded by a minus sign if the signal is deleted at that operator. Any signal which is not deleted is a final signal.

5.4 GO2

5.4.1 Kind Data

The kind deck contains input data for GO2 (the operator file, previously created by GO1, is also used). All operator types except 2, 10, and 11 require kind data. The deck contains (in this order):

a. Name Card
b. Kind Records
c. EOR Card

The contents of the name card become the kind file name. Column 80 of this card is reserved as a perfect case flag, i.e., if the character in column 80 is non-blank, the perfect case option will be selected (see Section 5.4.3).

Each kind record contains the data associated with the particular kind. The required data depends upon the operator type associated with the kind (see Chapter 6). The data are punched in the free-format manner described in Section 5.2.3.

5.4.2 Output

Besides creating the kind file for use by GO3, GO2 will produce the following printout:

a. The kind file name (the contents of the name card)
b. The date and time of the GO2 run.
c. A message that this is a perfect-case run if the flag value in column 80 on the name card is nonblank.
d. A listing of all kind records (including comments if present). Each record is preceded by the sequence number of the record. If GO3 detects an error in the data, an error listing message will be printed immediately below the record listing for that kind.

e. Additional error messages if inconsistencies between the kind data input and the kind data requested by the operator file are detected.

f. A summary table which lists (in order by kind number):
   1. kind number
   2. type associated with the kind
   3. frequency of the kind (the frequency value will be negative if the kind has been made perfect by use of the perfect case flag)

g. The number of kinds input.

h. The number of non-perfect kinds used.

i. The number of perfect kinds used.

In most instances item g will be equal to item h but it may be greater because it is not illegal to input extraneous kinds. This might well be the case where all kind data for a problem are input but a perfect-case run is requested. In this case all input kinds of types 1, 3, 6, 7, 16, and 17 would be extraneous.

5.4.3 Perfect Case Option

When this option is selected, GO2 automatically assigns 0 to the failure and premature mode probabilities and 1 to the normal (success) mode probability of all operators of types 1, 3, 6, 7, 16, and 17 as they occur in the operator file. This determines the operator, and consequently no user-supplied kind data is required for these types. Kind data for all other types (except 2, 10, and 11 which do not require kind data) must be supplied by the user as there is no reasonable way of automatically determining which of the possible operational modes is "perfect."

If any user-supplied kind data for types 1, 3, 5, 7, 16, or 17 is present, it will be checked for internal errors but will be ignored thereafter.

This option will normally be selected to assist the user in verifying the logical validity of the GO model and may profitably be used in conjunction with a complete intermediate distribution printout in GO3.

5.5. GO3

5.5.1 Analysis Deck

The analysis deck provides input data for GO3 (the operator and kind files, previously created by GO1 and GO2 respectively, are also used). There are several possible configurations for this deck.

The minimum configuration consists of just three cards: the name card, the parameter card, and an EOR card.

The name card is described in Section 5.2.1. Its contents will be printed as part of the heading. No other use is made of the information.

The parameter card format is described in Section 5.2.2. Each parameter name, function, possible value, and default value are given in Table 5.2. The default values are automatically assumed unless the user makes an explicit change on the parameter card.

If sensitivity runs are to be made, one sensitivity run subdeck must be used for each such run. These subdecks are all placed just after the parameter card. (See Section 5.5.3).

TABLE 3.2. GO3 PARAMETERS.

Parameter Name	Function	Possible Value	Default
PMIN	Probability cut-off for pruning	0.0 to 1.0	1.E-3
NEW	Omit regular run (get new data immediately)	0 means no 1 means yes	0 (no)
INTER	Print intermediate details	0 means no 1 means yes	0 (no)
SAVE	Save all distributions in the distribution file	0 means no 1 means yes	0 (no)
FIRST	Number of first operator for tracing	0-10000	10000
LAST	Number of last operator for tracing	0-10000	10000
TRACE	Cut-off probability for tracing	≥ 0.0	2.

5.5.2 Output

We first consider the case in which no sensitivity runs are made and no intermediate distribution information is desired (INTER = 0 and no tracing is requested). GO3 will then produce the following printout:

a. The analysis name.
b. The date and time of the GO3 run.
c. The creation date and time and the name of both the operator file and the kind file.
d. The value of infinity (largest possible signal value).
e. The maximum allowable distribution size.
f. The run number (incremented by 1 for each sensitivity run).
g. The values of the parameters PMIN, NEW, INTER, FIRST, LAST, and TRACE for the current run.
h. The Final Event Table which displays the final output distribution. For each term of this distribution the probability and the value of each of the final signals are given.
i. The Total Probability, which is the sum of all of the term probabilities in the final distribution.
j. The Total Error, which is 1 - Total Probability and is the sum of the probabilities of all terms which are pruned during the analysis by the PMIN criteria.
k. A table giving the marginal probability distribution of each final signal.

If the maximum array size is exceeded during the analysis of a particular operator, the value of PMIN is automatically increased and that operator reanalyzed (PMIN is multiplied by 10 unless it is initially zero, in which case it is set to $1.\times10^{-11}$). This increasing will be repeated

several times for an operator if necessary. PMIN is set back to its original value after the current operator analysis has been successfully completed. Every increase in PMIN will produce a message on the printout giving the operator number and the new value of PMIN.

If the value of the parameter INTER is 1, at the end of the analysis of each operator a printout of the operator number, its type and kind, and the size of the distribution produced by that operator will be made.

If the value of TRACE is less than or equal to 1.0, the printout at the end of the analysis of each operator will include that mentioned in the last paragraph and, if the operator number falls in the selection range established by the values of FIRST and LAST, a listing of each current distribution term whose probability is greater than or equal to the value of TRACE. The distribution term data will consist of the term probability followed by the number of each current signal and its value (in parentheses after the number). A signal number of zero means that no signal is currently occupying that position in the computer word in which the term is stored.

If sensitivity runs are made, each run will produce the same printout information as previously described except that the initial information (items a-e) will not be repeated, and the sensitivity run name and new kind data (if any) will be printed.

5.5.3 Sensitivity Runs

We will refer to an initial 303 run which makes exclusive use of the kind data in the kind file as a regular run

usually made for the purpose of investigating the sensitivity of the system's response to changes in one or more operator kinds).

Each sensitivity run requires its own subdeck. A sensitivity run subdeck is almost identical in makeup to the kind deck used for GO2. The only differences are:

a. A nonblank character in column 80 is a flag tellin GO3 that parameter changes are to be made and that the next card is a parameter card.

b. Data for only those kinds which are to be changed are included.

Thus a sensitivity run subdeck will contain either

a. name card (with a blank in column 80)

b. new kind data cards (optional)

c. EOR card,

or

a. name card (with a nonblank character in column 80)

b. parameter card

c. new kind data cards (optional)

d. EOR card.

During a GO3 analysis, as each operator is read from the operator file, its kind number is noted. If this kind was included in the new-kind subdeck for this analysis, the new data is used rather than the regular data in the kind file. An exception to this occurs if the selective operator option is used. This option permits the user to apply new kind data to only some of the operators of the particular kind, and is exercised by simply putting the number(s) of the operator(s) to which the new data is to apply on the new kind data record to the right of the kind data itself but before the terminating $. In addition, different new kind

data can be defined for different operator subsets of the same kind. As an example, assume operators 5, 13, 27, and 42 are all type 1, kind 99 operators. Assume that in the GO2 run, the kind success probability used was 0.9 - that is, the kind record used by GO2 was "99, 1, 0.9, 0.1 $". Then, if the new kind record(s) in the sensitivity run subdeck include

a. "99, 1, 0.5, 0.5 $", the success probability value of 0.5 will apply universally to all of the operators of kind 99 - that is, to operators 5, 13, 27 and 42.

b. "99, 1, 0.5, 0.5, 13 $", the probability value of 0.5 will apply only to operator 13 and the original value of 0.9 will apply to operators 5, 27, and 42.

c. "99, 1, 0.5, 0.5, 13, 42 $", and "99, 1, 0.3, 0.7, 5 $", the value of 0.5 will apply to operators 13 and 42, the value of 0.3 will apply to operator 5, and operator 27 will still use the original value of 0.9.

A universal (see a. above) new kind record and one for the same kind but using the selective operator option (b. or c. above) must not be present in the same sensitivity run; the results are unpredictible.

Several sensitivity runs may be made during a single GO3 run. The kind changes made in one sensitivity run are <u>not</u> carried along to subsequent runs, but all parameter changes except SAVE <u>are</u> unless new changes are explicitly made. The value of SAVE is always set to 0 at the end of every run.

The data in the kind file are not altered by a sensitivity run. The kind file itself can only be changed by rerunning GO2.

The value of the parameter NEW determines only if the first GO3 run is to be a regular run (NEW=0) or a sensitivity run (NEW=1). In the latter case, GO3 will first read the following sensitivity run subdeck and use the resulting new kind data, whereas in the former case, a regular run will first be made. In either case, after a run has been completed, GO3 will check to see if another sensitivity run subdeck is present; if it is, another sensitivity run will be made; if not, GO3 terminates.

Note that each sensitivity run subdeck must include a terminating EOR card.

Also note that if one or more sensitivity run subdecks are present, the last two cards of the analysis deck will be two EOR cards, the first terminating the last sensitivity run deck and the second terminating the analysis deck.

5.5.4 Tracing

The term "tracing" refers to the optional printing of some or all of the terms of some or all of the intermediate probability distributions. This is usually done for the purpose of checking or debugging the GO chart logic for a problem.

The particular operator distributions to which the trace option is to apply are defined by the values of FIRST and LAST. All distributions will be traced if FIRST=0 (or 1) and LAST = 10000 (or the largest operator number in the problem). No distributions will be traced if FIRST = 10000, LAST = 0, or FIRST>LAST. A single operator - say, number n - will be traced if FIRST = n and LAST = n.

If an operator distribution is selected for tracing, all terms of the distribution whose probabilities are equal to or greater than the value of TRACE will be printed as described in Section 5.5.2.

If a trace has been in effect during one run, it can be altered or turned off by appropriate parameter changes. Turning off can be accomplished by setting FIRST = 10000, LAST = 0, or TRACE > 1.0 or by any combination of these.

It is obvious that some care must be exercised in using the tracing option because an astronomical amount of print-out can easily be generated. We suggest that a fairly high value of TRACE - say, 0.5 - be used at first. This value will produce at most 2 terms per distribution because the sum of the term probabilities in any distribution cannot probabilities in any distribution cannot exceed 1.0 (and will equal 1.0 unless pruning has occurred).

We note that an initial run with no tracing but with INTER = 1 will give the size of each intermediate distribution, and this information can be used to establish reasonable tracing parameter values.

CHAPTER 6
GO OPERATOR TYPES

6.0 INTRODUCTION

In this chapter each regular GO type is discussed in detail. The discussion includes the exact data required for both the operator and the kind cards, the computational algorithms used by GO in processing an operator, and suggestions for use.

The algorithms should be studied carefully because they tell exactly what an operator does, and occasionally that is in conflict with the impressions that a casual user of GO may pick up from here and there. The computer programs themselves are, of course, the final source of definition information, but the statements given in this chapter are trustworthy.

The following symbols are used consistently, and others are defined as they occur:

- $S_1, S_2, \ldots$: the identification number of an input (Source, Stimulus) signal.
- $R_1, R_2, \ldots$: the identification number of an output (Result, Response) signal.
- K : the kind identification number.
- VS_i, VR_i : the value (time) of signal S_i or R_i.
- $P_1, P_2, \ldots$: probability of the described event.
- ∞ : infinity, the largest possible signal value (time).

When a type has only one input or output, the subscript on S or R will be deleted.

The operator and kind data are shown in the same order in which they must appear on the data cards. Data items have generally been separated by a comma and a blank, but any combination of blanks and/or commas is permitted. Each record must be terminated by a dollar sign ($) when it is punched, but the dollar signs are not shown here.

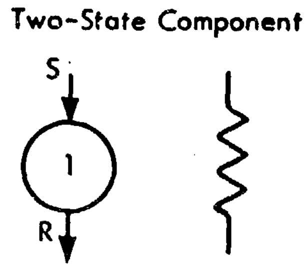
The sequence of regular operator data is: type, kind, number of inputs, inputs, number of outputs, outputs. The number of inputs and the number of outputs are omitted when the type definition requires a specific number--e.g., a type 1 always has one input and one output, and so these two 1's are not explicitly included in the data list.

Types 2 10 and 11 do not require kind data. The kind number in the operator data list is set to 0 for types 2 and 10 and is set equal to the value of the extra parameter for a type 11.

The kind data list starts out with the kind number and the type number, and these are followed by the required kind data.

6.1 Type 1: Two State Component

a. Symbolism:



b. Operator data: 1, K, S, R

c. Kind data: K, 1, P_1, P_2

P_1 Component is good

P_2 Component is bad (dud, failed)

d. Operation: VR = VS, if the component is good.

∞ = , if the component is bad.

e. Algorithm

VS	VR	Probability
∞	∞	1.
$<\infty$	VS ∞	P_1 P_2

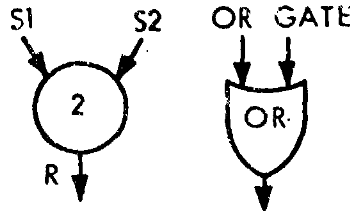
The type 1 component is perhaps the most common type. Any entity to which one ascribes two states can be modelled using this type. It was conceived originally to treat passive components like resistors or diodes which do not generate signals but only pass or fail to pass them. In preliminary studies many components or subsystems which could be modelled more precisely or accurately with other types are often represented as type 1 components to obtain quick results. In these cases timing considerations are generally ignored.

Assuming that it is important or desirable to account for time and operational sequences the following equipments would most likely be modeled as type 1 components.

Electrical		Mechanical
bell	heater element	joint
buzzer	lamp	elbow
annunciator	microphone	cross
horn	motor	reducer
amplifier	plug	lateral
antenna	receiver	stop cock
arrestor element	receptacle	safety valve
battery	rectifier	expansion joint
bimetal element	reproducer	union
bushing	resistor	sleeve
capacitor	resonator	brushing
circuit breaker	rheostat	straight pipe
coils	ringer	linkage
core	solder point	rod
crystal	sounder	piston
electrode	thermocouple	cylinder
fuse	transmitter	
ground	wiring	

6.2 Type 2: OR Gate

a. Symbolism:



b. Operator data: 2, 0, n, S,..., S_n, R

n = number of inputs, $2 \leq n \leq 10$

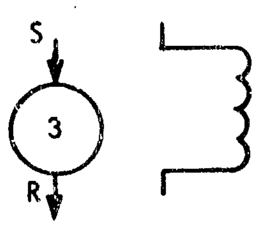
c. Kind data: None

d. Operation: $VR = \min \{VS_1, \ldots, VS_n\}$

The type 2 operator determines the smallest of its several input signal values. In many situations this value turns out to be associated with the connective "OR". In particular, when modeling a fault tree, the OR gate will be represented by type 2 operators.

6.3 Type 3: Triggered Generator

a. Symbolism: Triggered Generator



b. Operator data: 3, K, S, R

c. Kind data: K, 3, P_1, P_2, P_3

P_1: The generator is good.

P_2: The generator fails to operate

P_3: The generator operates prematurely.

d. Operation: VR = 0, if the generator prematures.

= , if the generator fails

= VS, otherwise

e. Algorithm

VS	VR	Probability
∞	0	P_3
	∞	P_1+P_2
0	0	P_1+P_3
	∞	P_2
>0 and	0	P_3
$<\infty$	VS	P_1
	∞	P_2

The type 3 operator is used to model various actuators which typically require inputs prior to generating an output but which may inadvertently produce an output signal in the absence of an input. This premature output may reflect improper reset conditions or responses to unexpected environmental stimuli - heat, shock, electromagnetic radiation, etc.

Typical devices which may manifest this behavior are switch and valve actuators.

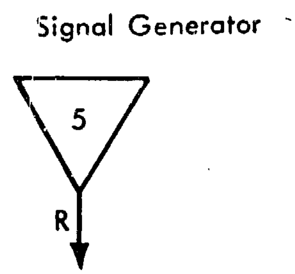
Many devices which might be modeled as type 3 operators are often modeled less accurately as type 1 operators when it is not desired to provide visibility of prematures. The type 3 is more general than a type 1. By creating a model using type 3 operators and setting the premature probabilities to zero, one has in effect used type 1 operators and has the flexibility to include prematures if desired, but at the expense of some computational inefficiency.

Relay coils, solenoids, actuating sensors, accelerometers, motors, pneumatic actuators, etc., can all be modeled using type 3 representations.

6.4 Type 4: This type is currently nonexistent.

6.5 Type 5: Signal Generator

a. Symbolism:



b. Operator data: 5, K, R

c. Kind data: K, 5, n, $V_1, P_1, \ldots, V_n, P_n$

n: number of signal values to be generated. $1 \leq n \leq 48$

V_i: i^{th} signal value

P_i: probability of the i^{th} value:

$$\sum_{i}^{n} P_i = 1.$$

d. Operation: $VR = V_i$ with probability P_i, $i = 1, \ldots, n$

The type 5 operator is typically used to generate values to initiate the sequence of operations and events modeled by a GO chart. Such initiating signals are often the outputs from other systems or subsystems not being explicitly addressed. Type 5 operators can also represent environmental effects like temperature or shock, extraneous RF signals, lightning, radiation, etc. The inclusion of independent actions by human operators are often introduced into the models with type 5 operators.

The flexibility afforded by the capability to generate events over the entire time or value spectrum provides an analytical convenience and a realistic representation of many random events induced outside the system being analyzed.

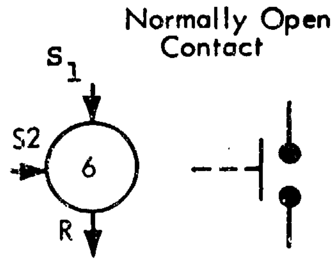
The most common use of the type 5 operator is to generate an initiating signal with just one value. When this is done, the kind data is registered as follows: K,5,1,V,1.0 where V is the value which is generated with certainty. It is sometimes convenient to designate on the GO chart the most likely value for the output signal.

The type 5 and type 13 operators are the only ones not requiring at least one input. Typically type 5 operators are represented with triangular symbols to identify the external system inputs. Because of the sequential nature of GO processing from operator to operator, it is axiomatic that every model must begin with at least one type 5 or 13 operator to initiate the sequence.

Note that two or more type 5 operators in a model will generate statistically independent signals. Dependent input signals can be modelled using a type 13 operator.

6.6 Type 6: Normally Open Contact

a. Symbolism:



b. Operator data: 6, K, S_1, S_2, R

c. Kind data: K, 6, P_1, P_2, P_3

P_1: contact operates normally

P_2: contact fails to close

P_3: contact closes prematurely

d. Operation:

$VR = \{\max VS_1, VS_2\}$, if contact operates normally.

$= VS_1$, if contact closes prematurely.

$= \infty$, if contact fails.

e. Algorithm

VS_1	VS_2	VR	Probability
∞	any	∞	1.
$<\infty$	∞	VS_1 ∞	P_3 P_1+P_2
	$\leq VS_1$	VS_1 ∞	P_1+P_3 P_2
	$>VS_1$ and $<\infty$	VS_1 VS_2 ∞	P_3 P_1 P_2

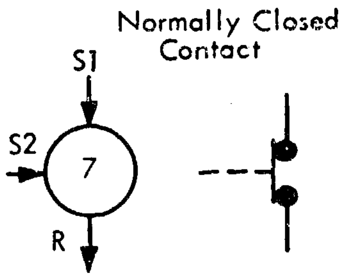
The type 6 operator is commonly used to represent electrical switches, fluid and pneumatic valves and certain logic event combinations. Signal passage on the main path is precluded until an auxiliary event closes the open gate. A type 6 operator with perfect probabilities of operation is an AND gate and could be interchangeable with a type 10 operator having two inputs.

Some common elements typically modeled using type 6 operators are:

normally open electrical switch contacts
normally disengaged clutch
normally disengaged gear
normally closed hydraulic valves

6.7 Type 7: Normally Closed Contact

a. Symbolism:



b. Operator data: 7, K, S_1, S_2, R

c. Kind data: K, 7, P_1, P_2, P_3

P_1: contact operates normally

P_2: contact fails to open

P_3: contact opens prematurely

d. Operation:

$VR = VS_1$, if the contact fails, or if $VS_2 \geq VS_1$ and the contact operates normally.

$= \infty$, otherwise.

e. Algorithm

VS_1	VS_2	VR	Probability
∞	any	∞	1.
$< \infty$	$< VS_1$	VS_1	P_2
		∞	P_1+P_3
	$\geq VS_1$	VS_1	P_1+P_2
		∞	P_3

Note the arbitrary convention that equal values of S_1 and S_2 cause R to take on the common value.

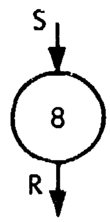
The type 7 operator is the reverse of the type 6 in that signal passage is possible until an auxiliary event occurs to open the gate and preclude passage. Its usage closely parallels that of the type 6 operator to model electrical switches, valves and logical event combinations.

Some elements typically modeled using type 7 operators are:

Normally closed electrical switch contacts
Normally engaged clutches
Normally open valves

6.8 Type 8: Increment Generator

a. Symbolism: Delay Generator



b. Operator data: 8, K, S, R

c. Kind data: K, 8, n, D_1, P_1,..., D_n, P_n

n: number of possible increments; $1 \leq n \leq 48$

D_i: value of the i^{th} increment, $-\infty \leq D_i \leq \infty$ (note: the D_i must be written in increasing order.)

P_i: Probability that the i^{th} increment occurs;

$$\sum_{i}^{n} P_i = 1.$$

d. Operation:

$VR=\max\left\{0, \min\left\{VS+D_i, \infty\right\}\right\}$ with probability P_i .

The type 8 operator is normally used to model delays in component responses. Timed sequences and operational lags are readily accounted for with type 8 operators. Typically a time increment of Dsignal is generated D_i time periods after receipt of the input with probability P_i.

Care should be exercised in using type 8 components because the delay D is measured in number of time points which may not correspond with clock time. That is, time periods generally are not of equal duration but are selected to denote significant system times of interest and are kept small in number to reduce the complexity. Hence, the user must carefully define his time line reference and insure that the delays defined are appropriate.

Some of the delays which are commonly modeled using type 8 components are:

Human operator responses

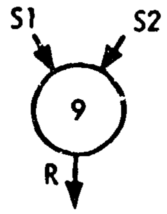
Mechanical response lags

Electrical system response lags

Timers and clocks

6.9 Type 9: Function Operator

a. Symbolism: Function Operator



b. Operator data: 9, K, S_1, S_2, R

c. Kind data: K, 9, n, X_1, Y_1,..., X_n, Y_n

n: number of X_i, Y_i pairs: $1 \leq n \leq 48$

X_i,Y_i: Signal values. The set of pairs defines Y_i as a function of X_i-i.e., Y_i=f(X_i). The value of X_i will be the difference VS_2-VS_1 between the values of the two input signals. Both X_i and Y_i may lie in the range from $-\infty$ to $+\infty$ inclusive. Values of X_i within that range which are not explicitly included in the kind data have an associated Y_i of $+\infty$.

d. Operation:

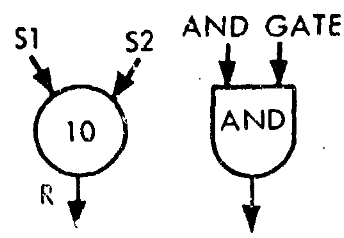
$$VR=\max\left\{0,\ \min\left\{VS_1 + f(VS_2-VS_1),\ \infty\right\}\right\}$$

The type 9 operator can be used to represent devices which require the arrival of two input signals in a specified sequence to produce an output. No physical device is suggested by the type 9, but it has proven convenient in many instances requiring synchronization.

In general, S_1 arrives first and S_2 arrives n time points later. The output signal R is generated f(n) time points after the arrival of S_1 where the discrete values of f(n) are provided as input data.

6.10 Type 10: AND Gate

a. Symbolism:



b Operator data: 10, 0, n, $S_1, \ldots, S_n$, R

n = number of inputs; $2 \leq n \leq 10$

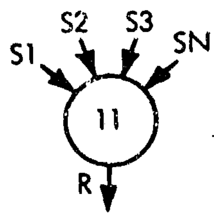
c. Kind data: none

d. Operation: $VR = \max\{VS_1, \ldots, VS_n\}$

A type 10 operator determines the largest of its several input signal values. In many situations this value turns out to be associated with the connective "AND". In particular, when modelling a fault tree, the AND gates will be represented by type 10 operators.

6.11 <u>Type 11:</u> <u>m out of n Gate</u>

a. Symbolism: M out of N Gate



b. Operator data: 11, m, n, $S_1,\ldots, S_n$, R

n= number of inputs; $2 \le n \le 10$

m= gate parameter; $1 \le m \le n$

c. Kind data: none

d. Operation:

Let V_1, $V_2,\ldots$, V_n be the ordered set of values of VS_1, $VS_2,\ldots$, VS_n (from smallest to largest).

Then: $VR=V_m$

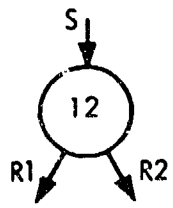
e. Comments:

1. Note that the kind number in the operator data is replaced with the gate parameter.
2. If m=1, this type is equivalent to a type 2; and if m=n, it is equivalent to a type 10.

A type 11 operator could be replaced with appropriate combinations of type 2 and 10 operators, but use of a type 11 simplifies modelling the logical combination of more than two signals.

6.12 Type 12: Path Splitter

a. Symbolism: Path Splitter



b. Operator data: 12, K, S, m, $R_1, \ldots, R_m$

m = number of outputs, $1 \le m \le 10$

c. Kind data: K, 12, m, $P_1, \ldots, P_m$

P_i = probability that i^{th} path is "selected";

$$\sum_{i=1}^{m} P_i \le 1.0$$

d. Operation:

m+1 terms are produced. The first m of these are defined by:

$VR_i = VS$ and $VR_j = \infty$ for all $j \neq i$, with probability P_i, $i=1, \ldots, m$

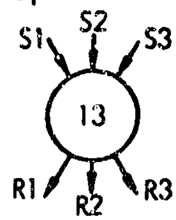
and the $m+1^{st}$ is defined by

$VR_j = \infty$ for all j, with probability $1 - \sum_{1}^{m} P_i$

(The $m+1^{st}$ term does not occur if $\sum P_i = 1$.)

6.13 Type 13: General Purpose Multiple Input, Multiple Output Operator

a. Symbolism: Multiple Input/Output Operator



b. Operator data: 13, K, n, $S_1, \ldots, S_n$, m, $R_1, \ldots, R_m$

n = number of inputs; $0 \le n \le 10$

m = number of outputs; $1 \le m \le 10$

c. Kind data: K, 13, n, m, N

$VS_{11} \ldots VS_{n1}\ M_1$

$VR_1 \ldots VR_m\ P_{11}$

$\vdots$

$VR_1 \ldots VR_m\ P_{M11}$

$\vdots$

$VS_{1N} \ldots VS_{nN}\ M_N$

$VR_1 \ldots VR_m\ P_{1N}$

$\vdots$

$VR_1 \ldots VR_m\ P_{M_nN}$

where:

n = number of inputs; $0 \leq n \leq 10$.

m = number of outputs; $1 \leq m \leq 10$.

N = number of output sets; $N \geq 1$. (if $n=0$, $N=1$).

M_i = number of output terms for the i^{th} output set.

$VS_{1i}, \ldots, VS_{ni}$: the i^{th} input value comparison set (missing if $n=0$).

P_{ij} = probability of the i^{th} output term in the j^{th} output set:

$$\sum_{i=1}^{M_i} P_{ij} = 1,\ j = 1, \ldots, N$$

c. Operation:

If $n > 0$, the actual input values are compared with the N input value comparison sets. If a match is found, the corresponding joint output distribution is produced. If no match is found, all output values are set to ∞ (with probability 1).

If $n = 0$, the single joint output distribution is produced.

d. Comments:

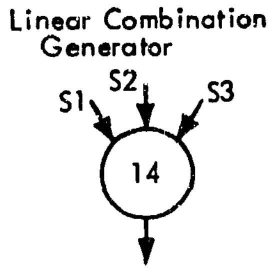
1. The maximum amount of kind data is limited to 100 data items.
2. For legibility the kind data should probably be above rather than simply strung out item-by-item.
3. In principle, any of the other GO types could be replaced by a properly defined type 13. However, the amount of kind data required for a complete definition is prohibitive in most cases.
4. Setting n (# of inputs) to zero gives us a signal generator which can produce several dependent signals (as contrasted to independent signals which would be produced by several type 5's).

5. A type 13 can be easily used as a nonstochastic function device in which a single (multiple) output is defined as a function of a single (multiple) input.

The type 13 operator provides for complete generality in defining how an equipment or subsystem behaves. Because of its complexity, however, its use is severely restricted.

6.14 Type 14: Linear Combination Generator

a. Symbolism:



b. Operator data: 14, K, n, $S_1, \ldots, S_n$, R

n = number of inputs; $1 \leq n \leq 10$

c. Kind data: K, 14, n, $a_1, \ldots, a_n$, a_o

a_i: any real number

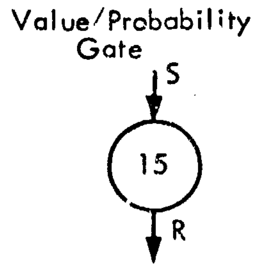
d. Operation:

Let A be the value of $a_0 + a_1\ VS_1 + \ldots + a_n\ VS_n$ rounded to the nearest integer. Then

$$VR = \max\{0, \min\{A, \infty\}\}$$

6.15 Type 15: Value/Probability Gate-Generator

a. Symbolism:



b. Operator data: 15, K, S, R

c. Kind data: K, 15, V_1, V_2, V_3, V_4, P_1, P_2

V_1 = output value if input is in gate. (set to -1 if output value is to equal input value).

V_2 = output value if input is not in gate.

V_3, V_4 = value gate values; $0 \leq V_3 \leq V_4 \leq \infty$.

P_1, P_2 = probability gate values: $0 \leq P_1 \leq P_2 \leq 1$.

d. Operation:

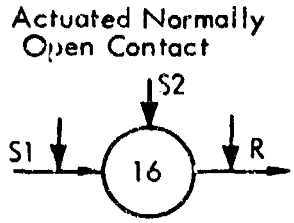
Let $V = V_1$ if $V_1 \geq 0$
$= VS$ if $V_1 = -1$

and PS = probability associated with the input term.

Then $VR = V$, if $V_3 \leq VS \leq V_4$ and $P_1 \leq PS \leq P_2$
$= V_2$, otherwise.

6.16 Type 16: Actuated Normally Open Contact

a. Symbolism:



b. Operator Data: 16, K, S_1, S_2, R

c. Kind data: K, 16, P_1, P_2, P_3

P_1: contact operates normally

P_2: contact fails (fails open)

P_3: contact prematures (fails closed)

d. Operation:

$$VR = 0 \text{ , if contact fails open}$$
$$= VS_1 \text{ , if contact fails closed}$$
$$= \min \{VS_1, VS_2\} \text{ , if contact operates normally}$$

e. Algorithm

VS_1	VS_2	VR	Prob.
0	any	0	1.
>0	$\leq VS_1$	0 VS_1 VS_2	P_2 P_3 P_1
> 0	$>VS_1$	0 VS_1	P_2 P_3+P_1

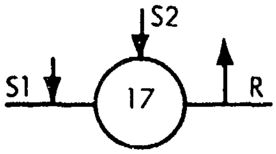
The type 16 operator is different from all those previously discussed in the conceptual nature of its inputs and outputs. With the exception of type 17 all other operators are conceived to have input and output signals which are arriving, that is, going from an off to an on condition. The type 16 operator is defined to address cases where the events of interest constitute signal termination. This situation frequently occurs in systems involving alarm devices or subsystems - for example, a scram system in a power plant.

The type 16 operator can be conceived as a normally open switch contact being held in its closed or actuated position (an actuated type 6). Its primary input is an on-to-off signal, i.e., one which terminates. The secondary input could be either an on-to-off or an off-to-on signal. The generated output is an on-to-off signal modeling the cessation of power or termination of the event represented.

6.17 Type 17: Actuated Normally Closed Contact

a. Symbolism:

Actuated Normally Closed Contact



b. Operation data: 17, K, S_1, S_2, R

c. Kind data: K, 17, P_1, P_2, P_3

P_1: contact operates normally

P_2: contact fails (fails closed)

P_3: contact prematures (fails open)

d. Operation:

VR = , if (1) contact normal and $VS_2 \geq V$

(2) contact fails open

(3) contact fails closed and $VS_1 = 0$.

= 0, if contact fails closed and $VS_1 > 0$.

= VS_2, if $VS_1 > 0 = VS_2 < VS_1$m and contact normal

Note the arbitrary convention that if $VS_1 = VS_2$, VR =

e. Algorithm

VS_1	VS_2	VR	Probability
0	any	∞	1.
> 0	$< VS_1$	0	P_2
		∞	P_3
		VS_2	P_1
	$\geq VS_1$	0	P_2
		∞	P_3+P_1

The type 17 operator is similar to the type 16 operator in that the primary input signal is an on-to-off signal. Conceptually it is an actuated normally closed contact, i.e., one held in an open position (an actuated type 7). It has been defined to examine events of interest where signal cessation instead of signal arrival is being addressed.

The primary input signal is an on-to-off signal, the secondary input could be either an on-to-off or an off-to-on signal. The generated output signal is an off-to-on signal.

CHAPTER 7
SUPERTYPES

7.0 INTRODUCTION

A supertype is a structured collection of operators which the analyst chooses to treat as a single entity. This will normally be done when the collection represents a physical subsystem of the modeled system which either occurs several times within the system or when the understanding of the model is enhanced by viewing it on a subsystem rather than a component basis.

When repeated subsystems occur, a considerable saving of analyst time and effort is achieved by using supertypes because the fundamental structure of the subsystem need be modeled only once rather than at each occurrence.

Even if no subsystem repetition occurs, the GO model can frequently be greatly improved from the viewpoint of the original analyst and others who will use the modeling results by the judicious use of supertypes. Basically, supertypes permit a "block-diagram" approach to a model; and, because nested supertypes are permitted, a heirarchy of diagramming levels is possible.

It must be emphasized that the use of supertypes is strictly for the convenience of the analyst because all supertypes will be translated into their constituent regular operators by GO1 before the operator file is created. Consequently, no computer execution time is saved.

In the remainder of this chapter we will illustrate the mechanics of defining and using supertypes by means of a single example. Several illustrations of their use will be found in Chapter 11 and a brief summary of the usage rules in Appendix D2.

7.1 Example

Let the GO chart of a simple system be that shown in Figure 7.1.

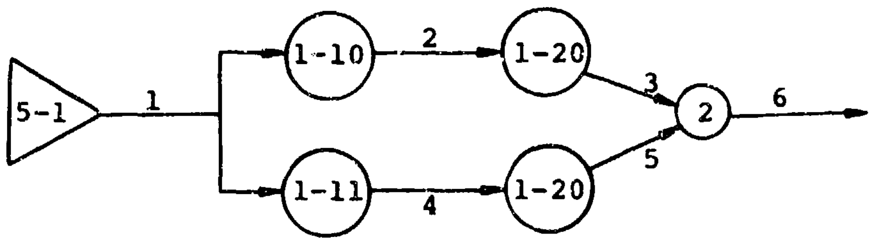


FIGURE 7.1. INITIAL SYSTEM GO CHART.

We will first create a supertype consisting of two type 1 operators in series and use it to model the two such subsystems in Figure 7.1. The GO chart for the supertype is shown in Figure 7.2. (The dashed rectangle indicates the boundaries of the supertype).

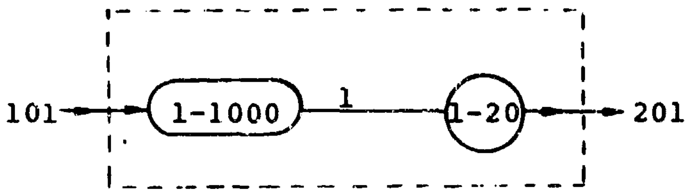


FIGURE 7.2. SUPERTYPE 100 DEFINITION.

Let us pause and explain the notation and numbering schemes.

The type number of a supertype is assigned by the user and must be greater than 99. In the definition - e.g., Figure 7.2 - the supertype input signal numbers must be in the range 100 to 199 inclusive and the output signal numbers 200 to 299. The internal signals - that is, those signals which are neither inputs nor outputs to or from the supertype - must be numbered between 1 and 99 inclusive and should preferably be numbered sequentially starting at 1.

Note that all signal numbers within a supertype definition are "local" in nature - that is, they pertain only to the definition - and consequently they may duplicate numbers used in other supertype definitions or in the main model itself.

A kind number of a component within a supertype definition can be either <u>fixed</u> or <u>variable</u>. It is fixed if it is not going to be different in different uses of the supertype and variable otherwise. In Figure 7.2 the kind number 20 associated with the second type 1 operator is fixed because inspection of Figure 7.1 shows that it will be the same in both applications of the supertype. On the other hand, the kind of the first type 1 is variable because it will be 10 in one application and 11 in the other. Within the supertype definition variable kinds are given values greater than 999 while fixed kinds are assigned the usual values (from 1 to 999).

The GO1 operator deck data cards which will define supertype 100 are:

```
100  -1 1000 101 201 $
1  1000  101  1  $
1  20  1  201  $
EOR
```

The first card - the <u>supertype declaration card</u> - contains the supertype number (100), -1 (which is simply a flag for use by GO1 and indicates that this is a supertype definition rather than a supertype use), the variable kind number (1000), the input signal number (101), and the output signal number (201). The next two cards are the regular operator cards which actually define the supertype, and the definition is terminated by an EOR card.

The kind and signal numbers appearing on the declaration card and following the -1 flag constitute the argument list for the supertype. Up to 25 entries are permitted in the argument list, and they may be any combination of variable kind numbers and input or output signal numbers. Their order is immaterial but must be consistent between the definition and the use of the definition in a super operator (see below).

The use of our supertype is shown in Figure 7.3 in which we have replaced both occurrences of the series type 1's by the supertype.

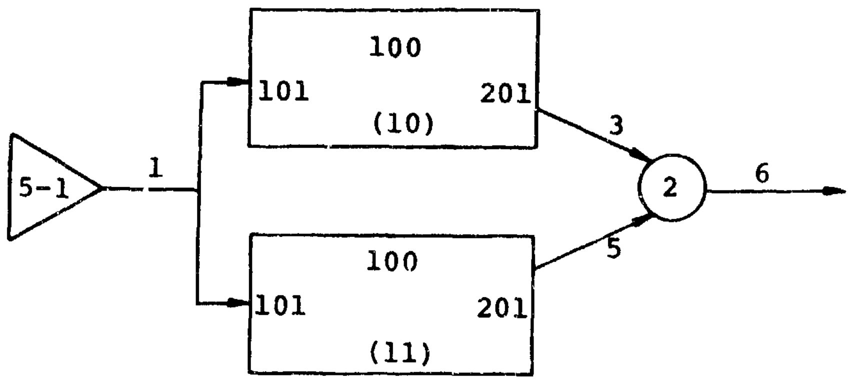


FIGURE 7.3. SYSTEM WITH SUPERTYPE 100.

The two super-operators are shown by rectangles, and the numbers inside the rectangle identify the various arguments associated with the supertype. The "101" and "201" are not actually needed in this case because there is just a single input and a single output. When a supertype has several inputs or outputs, identification such as indicated is necessary. The "10" and "11" in parentheses are the actual kind numbers which are to be identified with the variable kind 1000 in the definition.

The data cards for our system model will be as follows:

```
5   1   1   $
100   0   10   1   3   $
100   0   11   1   5   $
2   0   2   3   5   6   $
```

The second and third cards are the two super-operator cards that is, the two uses of the supertype. The data on these are the supertype number (100), the bias control number (0-see below), and the system numbers which are to replace the definition numbers associated with the variable kind, input signal, and output signal (these numbers must correspond with those in the argument list of the supertype declaration card).

When G01 encounters the first super-operator, it will use the supertype 100 definition to create a set of regular operators. The numbers associated with those operators are obtained by replacing the numbers in the definition as follows:

(1) Variable kind numbers and supertype input and output signal numbers are replaced by the corresponding numbers given on the super-operator card. Thus, 1000 is replaced by 10, 101 by 1, and 201 by 3.

(2) Fixed kind numbers are left alone. Thus 20 remains 20.

(3) Internal signal numbers are replaced by either the sum of the definition internal signal number and the G01 parameter BIAS (default value = 5000) or the sum of the internal signal number and the bias control value on the super-operator card if the bias control value is greater than 0. In our example the internal signal number is replaced by 1+5000 = 5001 because the bias control is 0 (it must always be nonnegative).

As a result of these replacements, G01 will create the following operators which will be used in the actual system model to be analyzed by G03:

1 10 1 5001

1 20 5001 3

These operators will be shown on the G01 printout following the super-operator card.

In a similar manner the second super-operator will result in

1 11 3 5002

1 20 5002 5

The internal signal number is 5002 rather than 5001 because in effect the value of the parameter BIAS is increased to the largest replaced internal signal number of the previous super-operator. Thus, after the first super-operator had been processed, the effective value of BIAS had been set to 5001.

The use of BIAS and the bias control values on the super-operator cards can become very confusing. The goal is to produce in the final system model signal numbers which are not duplicated and which do not exceed the maximum permitted value of 9999. We suggest that the value of BIAS be left at its default value of 5000, that 0 be used as the value for <u>all</u> bias controls, and that all regular signal numbers in the model be restricted to values under 5000. If this is done, the user can anticipate no difficulties unless the overall model becomes excessively large.

7.2 Supertype Nesting

To illustrate the supertype nesting concept, let us take the previous example and define another supertype which

will include the type 5 and the two type 100's. This super-type let's call it type 110 - will have no inputs, two outputs, and no variable kinds. The GO chart is shown in Figure 7.4.

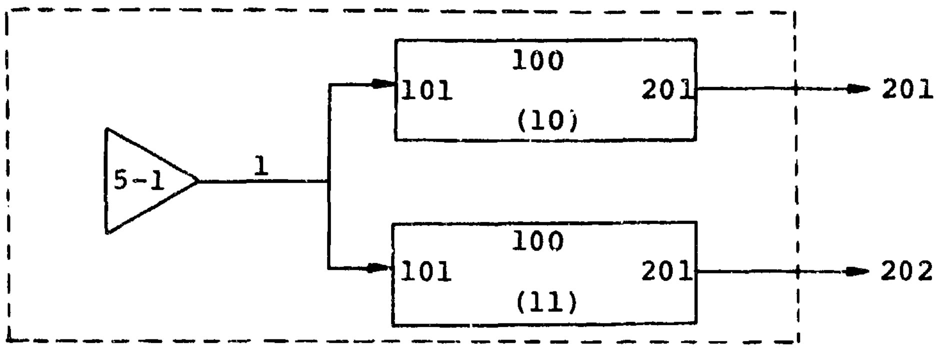


FIGURE 7.4. SUPERTYPE 110 DEFINITION.

The data cards required for the definition are:

```
110  1  201  202  $
5  1  1  $
100  0  10  1  201  $
100  0  11  1  202  $
EOR
```

The system GO chart is:

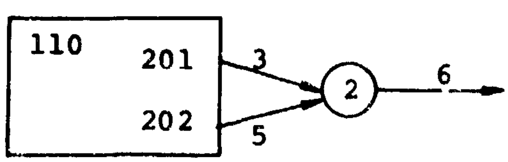


FIGURE 7.5 SYSTEM WITH SUPERTYPE 110.

and the required data cards are

```
110  0  3  5  $
2  0  2  3  5  6  $
```

When GO1 processes the type 110 super operator, the following operators will be produced:

```
5  1  5001
1  10  5001  5002
1  20  5002  3
1  11  5001  5003
1  20  5003  5
```

In the GO1 printout of these operators, the type 5 will be preceded by "L=1" and the other four by "L=2". L refers to the nesting level of the supertype which created the operator. Nesting up to five levels is permitted.

We note that the data submitted to GO1 would include the definition of supertype 100, the definition of supertype 110, and the system definition. The supertype definitions must precede the system definition. As a general rule we suggest that all supertype definitions be placed at the beginning of the operator deck (following the GO1 parameter card). The order in which they are placed is immaterial even if nesting is involved.

7.3 Miscellaneous Comments

1. Up to 20 supertypes may be defined in a GO model.
2. A supertype may not be nested within itself either directly or indirectly (doing so will produce a "NESTED TOO DEEP" error message).
3. The argument list must contain at least one entry.

4. The total amount of data in all supertype definitions cannot exceed about 4000.
5. An internal super-operator signal may be a final output signal. In most cases a trial run of GO1 will be needed to determine the number of such a signal.

CHAPTER 8
PROGRAM RESTRICTIONS

8.0 INTRODUCTION

The following restrictions must be observed. Failure to do so will usually result in an error diagnostic, but unpredictable results may occur. Any of the restrictions which involve a numerical limitation can be altered (with considerable difficulty in some cases) by appropriate modifications to the programs; see the "GO Program Manual" for details.

a. Operator data:

1. Signal numbers must be unique - that is, the same number cannot be used for different signals (except within different supertype definitions).
2. The maximum signal number (and consequently the maximum number of signals) is 9999.
3. The number of signal values must be between 2 and 128 inclusive.
4. Let M be the maximum number of simultaneously active signals, and let N be the number of signal values. Then M is a function of N as shown in the table below.

N	M
1 to 2	300
3 to 4	150
5 to 8	100
9 to 16	75
17 to 32	60
33 to 64	50
65 to 128	40

The actual number of signals is determined in G01 and is included in the printout from G01. If this value exceeds the maximum number permitted, the user may (a) reorder the signals (b) decrease the number of signal values, and/or (c) increase the maximum number allowed (a nontrivial reprogramming chore).

5. The maximum number of final signals is 24.
6. The maximum number of signal inputs for a regular multiple-input operator is 10.
7. The maximum number of signal outputs for a regular multiple-output operator is 10.
8. The maximum number of supertypes defined is 20. There is no limit to the number of super-operators created from the supertypes.
9. Supertypes may be nested up to 5 deep.
10. The maximum total amount of supertype data is 2000.
11. The total number of arguments (input and output signal numbers and kind numbers) on a supertype declaration card can be no greater than 25. There is no other restriction on the number of input or output signals - for example, a supertype may have 3 inputs and 22 outputs.

b. Kind data:

1. Each kind is associated with a single operator type. Thus, the same kind may not be used with different types even though the kind data are compatible.
2. The largest kind number (and consequently the maximum number of kinds) is 999. (Note: the

kind numbers included in the argument list of a supertype are "dummy" values and must be greater than 999).

3. The maximum amount of data in a single kind input record is 100.
4. The total amount of kind data for GO2 cannot exceed 700 integer and 700 noninteger numbers.
5. The total amount of new kind data for a sensitivity run in GO3 cannot exceed 500 integer and 500 noninteger numbers.

c. Other:

1. The maximum number of computer words used for holding the signal values for each term of the probability distributions is 5. The number actually used is determined by GO1 and depends upon the number of signal values and the maximum number of active signals.
2. The maximum distribution size is 3000 divided by the number of words per term.
3. The maximum amount of selective operator data in a sensitivity run is 50.

PRECEDING PAGE BLANK-NOT FILMED

CHAPTER 9
ERRORS

9.0 INTRODUCTION

A substantial amount of checking for errors is done by the GO programs. In most cases one or more errors detected by a program will result in appropriate error messages, but in some unanticipated situations the results are unpredictable.

As a general rule errors in, say, GO1 will cause the entire run to be aborted - that is, GO2 and GO3 will not be executed even though they were requested. There may be exceptions to this.

The three sections of this chapter list the possible error messages produced by the three GO programs and provide some supplemental information about most of them.

9.1 GO1 Errors

GO1 makes numerous checks on the validity and consistency of the input data. If an error is detected, an appropriate error message is printed, and in most cases, after the preliminary checking of the entire operator deck has been completed, the message "SUICIDE BECAUSE OF ERRORS" will be printed, and a fatal operation performed. (This stopping method prevents any attempt to execute GO2 if it is included in the same computer run).

In some cases a single error will produce several error messages because of a cumulative effect. Also, only one error message relating to an operator will be issued even though there might be several errors associated with the operator data.

Data for operators contained within a supertype receive only cursory checks when the supertype is defined. Consequently, most errors in such data will not be revealed until the supertype is actually used as a super-operator.

Appendix B includes the results of a G01 run in which numerous errors were intentionally made. Except for the introduced errors, the operator deck is the same as that for the main example in the Appendix.

Fatal error messages are:

a. SIGNAL s IS OUT OF RANGE

   A signal number must lie between 1 and 9999 inclusive.

b. SIGNAL s REUSED

   Signal numbers must be unique.

c. INPUT SIGNAL s HAS NOT BEEN ENTERED

   A signal must occur as an operator output before it can be used as an input. When this error occurs, the missing signal number will be artificially supplied to prevent propagation of the error.

d. WRONG AMOUNT OF DATA

   There is either too little or too much data in the operator record. This is frequently caused by failing to put the terminating $ at the end of the previous record.

e. BAD PARAMETER

   Check the parameter restrictions for the operator type.

f. FINAL SIGNAL s OUT OF RANGE

g. FINAL SIGNAL s WAS NEVER ENTERED

h. FINAL SIGNAL CARD MISSING

i. THERE ARE TOO MANY FINAL SIGNALS

   24 is the upper limit

j. SIGNAL s, OPERATOR 0, OVERFLOWED SIGNAL LIST
There are too many concurrently active signals (See Chapter 8).

k. VALUES IS OUT OF RANGE
The number must lie between 2 and 128 inclusive.

l. NUMBER OF DATA EXCEEDS LIMIT OF 50

m. DATA ITEM EXCEEDS 4 CHARACTERS
9999 is the largest possible value for any data item.

n. TOO MUCH SUPERTYPE DATA
See Chapter 8.

o. UNDEFINED SUPERTYPE

p. TOO MANY SUPERTYPES
The limit is 60.

q. SUPEROPS NESTED TOO DEEP
The limit is 5.

r. SIGNAL NUMBER OVERFLOW
Signal numbers generated internally in super-operators exceed 9999. Adjust the bias.

s. KIND k NOT IN LIST. REPLACED BY 999.
A kind number greater than 999 is present in the operators of a supertype definition but is not included in the declaration list. The temporary replacement prevents error propagation.

t. KIND OUT OF RANGE

u. TOO MUCH DECLARATION DATA
Only 25 items are permitted. The list will be trimmed to 25, and some errors will probably occur later as a result.

If the cumulative number of errors exceeds the value of ERRORS (default = 25), the execution of GO1 will be stopped.

Inclusion of any nonnumeric character (other than a blank or comma) in the data field of an operator record will cause an instantaneously fatal format error.

9.2 GO2 Errors

Error checking is done in two stages: internal, in which the kind records themselves are checked, and external, in which the consistency between the kind data and the operator data is checked. Any detected error is fatal and will eventually produce a suicide message and action. Internal checking is performed first, and if an error is found, external checking is not done.

Internal error messages include:

a. NUMBER OF DATA EXCEEDS LIMIT OF 100

b. DATA ITEM EXCEEDS 10 CHARACTERS

c. v IS A BAD VALUE
   signal value is less than 0 or greater than infinity.

d. p IS A BAD PROBABILITY
   A data item which is supposed to represent a probability falls outside the range 0 to 1.

e. PROBABILITY SUM IS p
   Several probabilities which are supposed to sum to 1 do not (actual sum is p).

f. d IS OUT OF RANGE
   A type 9 data item is either greater than infinity or less than negative infinity.

g. KIND DUPLICATION
   Kind numbers must be unique.

h. WRONG AMOUNT OF DATA IN RECORD
   Either too much or too little.

i. TOO MUCH DATA. SUICIDE IS THE ONLY WAY OUT.
   The total amount of kind data entered is too large. This error produces an immediate suicide.

Most internal errors will be due to leaving out one or more required data items. Check the kind format in Chapter 6 carefully. Also, note that if an error is detected in a record, no further checking of that record is done.

External error messages include:

a. INCONSISTENT PARAMETER. OP o, KIND k, TYPE t. This will usually refer to an inconsistency in either the number of inputs or the number of outputs.

b. OP o NEEDS NONEXISTENT KIND k

c. OP o SAYS KIND k IS FOR TYPE t
Each kind must be associated with a unique type.

9.3 GO3 Errors

The following error messages may occur:

a. DISTRIBUTION HAS VANISHED AT OPERATOR o.
This occurs if all terms of the current probability distribution have been eliminated because their probabilities are all less than or equal to PMIN. The run will immediately terminate. The remedy is to decrease the size of PMIN.

b. KIND FILE NEEDS OP FILE name (date time).
The kind file contains the name and creation date and time of the operator file which was used by GO2 when the kind file was created. If this operator file creation data does not agree with the creation date of the operator file being used by GO3, this message will be issued and the run terminated.

c. NEWOP OVERFLOW
The total amount of selective operator data in a sensitivity run is excessive.

d. OPERATOR o OUT OF RANGE. FATAL.
The operator number used in a selective operator option or a new kind card is less than 1 or greater than the number of operators in the problem.

During a sensitivity analysis, all new kind data is checked for internal consistency just as in GO2, and any of the internal check error messages may appear. In addition, consistency checks between the new kind data and the corresponding operator data may produce either of the following messages.

a. OP o, TYPE t1, USES KIND k, BUT NEW KIND IS FOR TYPE t2.

b. BAD PARAMETER IN KIND k, OP o, TYPE t.
OLD VALUE = x, NEW VALUE = y.

CHAPTER 10
HELPFUL HINTS

10.0 INTRODUCTION

This chapter contains a number of hints which may be of some help to a user in creating and running a GO model. Included are suggestions which have not been touched upon in other chapters or which have been mentioned in one form or another but deserve extra emphasis. Most of these hints concern matters which have arisen during the several years of GO's existence and which have frequently been bothersome.

10.1 Infinity

Try to make infinity as small as possible in order to get as many active signal values as possible into one computer word. In particular, try to make infinity equal to or less than one of 1, 3, 7, 15, 31, or 63. The point is to keep the maximum allowable distribution size in GO3 as large as possible, and this is accomplished by keeping all of the active signal values in one word.

10.2 Active Signals

When modeling, try to keep the number of active signals as small as possible. This will help keep the actual distribution sizes small and hence provide greater accuracy by reducing the PMIN pruning. This is normally done by the sequence in which operators are introduced which determines when signals are generated and deleted.

10.3 GO Chart

1. Draw a good GO chart - several tries will probably be needed.

2. Keep the model as simple as possible.

3. Use supertypes whenever they are appropriate.

10.4 PMIN

Start out with a relatively large value of PMIN and then decrease it on later runs. This will be faster (and cheaper) than going the other way.

10.5 Operator Sequencing

The sequence in which operators are processed can be crucial in complex models. A poorly chosen sequence can cause excessive pruning with a resulting high total error because of an unnecessarily large number of active signals.

-126-

d. OPERATOR o OUT OF RANGE. FATAL.
   The operator number used in a selective operator option or a new kind card is less than 1 or greater than the number of operators in the problem.

During a sensitivity analysis, all new kind data is checked for internal consistency just as in GO2, and any of the internal check error messages may appear. In addition, consistency checks between the new kind data and the corresponding operator data may produce either of the following messages.

a. OP o, TYPE t1, USES KIND k, BUT NEW KIND IS FOR TYPE t2.

b. BAD PARAMETER IN KIND k, OP o, TYPE t.
   OLD VALUE = x, NEW VALUE = y.

CHAPTER 11
EXAMPLES

11.0 INTRODUCTION

The following example applications of the GO modeling and analysis procedure have been selected to portray certain capabilities and provide a reference of typical usages. The cases chosen for inclusion here have been relatively simple and illustrative of certain conventions and characteristics of the GO procedure.

The following table lists the standardized operator types used in the GO models for each example:

TABLE 11.1. GO OPERATOR TYPES APPLIED.

EXAMPLE NO.	OPERATOR TYPES USED
1	1,5,11
2	1,2,5,11
3	1,2,3,5,6,9,10
4	2,5,10
5	1,3,5,16,17
6	1,2,5,10
7	1,2,5,6,7,8,9
8	1,2,5
9	1,2,5,6,10
10	1,2,3,5,6,7,8,9,10,13

In most instances the examples have been extracted from prior publications, reworked and rewritten for the present purpose. In each instance, the principal reference is noted beside the title of the example.

Only selected portions of the input and output data are shown. We suggest that, if the reader has access to the GO programs, running one or more of these examples will provide good practice in the mechanics of using the programs.

11.1 Example 1 - Network #1 of a Power Distribution System [8]

The purpose of this example is to portray the use of the GO methodology to analyze a simple two-out-of-four system.

Assume that the following elements of a power distribution system are configured such that successful electrical power transmission requires at least two of the four identical elements to be functional. Presume that it is additionally known that the failure distributions for these elements are exponential and that the failure rate, λ, is 0.48×10^{-6}/hr.

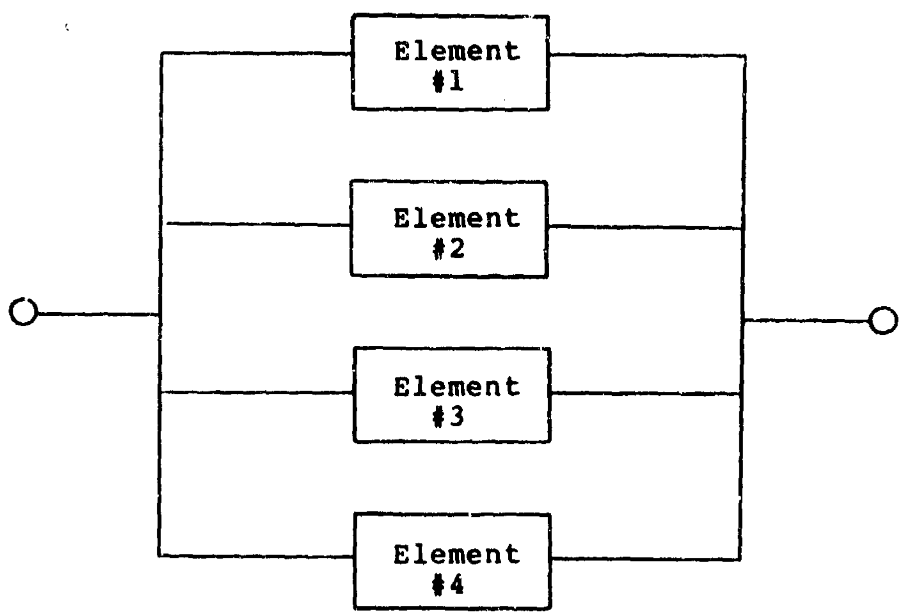


FIGURE 11.1.1. NETWORK #1 OF A POWER DISTRIBUTION SYSTEM.

It is desired to determine the probability that the network is still functional after 50,000 hours given that no repairs have been effected.

Figure 11.1.2, GO Reliability Diagram of Network #1, portrays the use of a type 5 operator to represent the input to this subsystem, the use of type 1 operators to represent the elements and the type 11 operator to combine the outputs of the four elements such that the subsystem is operational if at least two of the elements are functional.

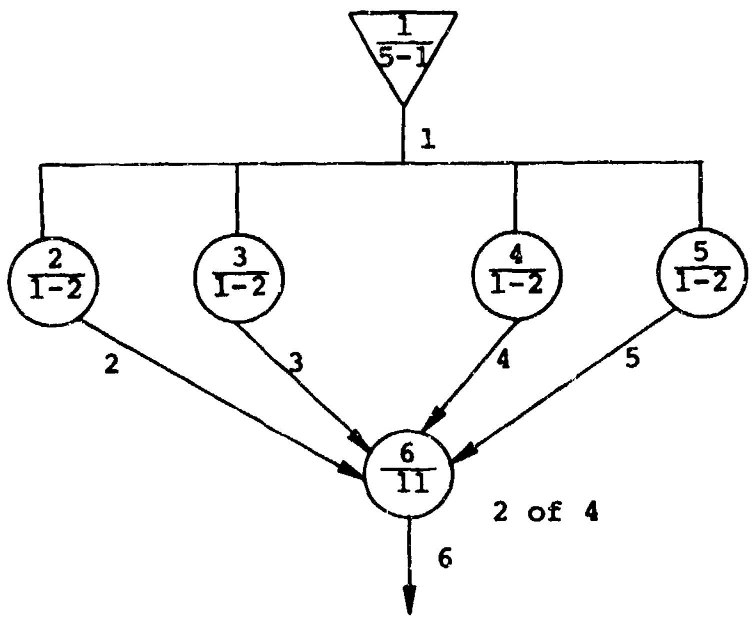


FIGURE 11.1.2. GO RELIABILITY DIAGRAM OF NETWORK #1.

The GO input data deck depicting the actual card images is shown below in Table 11.1.1.

TABLE 11.1.1. GO OPERATOR-KIND PARAMETER DATA FOR EXAMPLE 1

```
GO APPLICATIONS EXAMPLE - NETWORK 1
$PARAM VALUES=2 $
5 1 1$
1 2 1 2$
1 2 1 3$
1 2 1 4$
1 2 1 5$
11 2 4 2 3 4 5 6$
0 6 $
EOR
GO APPLICATIONS EXAMPLE 1 - NETWORK 1
1,5,1,0,1.$
2,1,0.9762858,0.0237142$
EOR
GO APPLICATIONS EXAMPLE 1 - NETWORK 1
 $PARAM PMIN=1.E-10$
EOR
```

One might verbalize this input data in this manner. Operator type 5 has kind 1 probabilities of operation and produces signal number 1. Operator type 1 has kind 2 probabilities of operation, has signal 1 as an input and generates signal 2, etc. Operator type 11 is a 2 out of 4 gate with inputs 2,3,4 and 5 and generates output signal 6.

Kind 1 documents the probabilities for the type 5 operator which generates a single output at time point 0 with probability 1.0. Kind 2 documents the probabilities for the type 1 operators (all of which are identical) which are 0.9762858 for success and 0.0237142 for failure after 50,000 hours of operation. These kind data values were determined by noting that the reliability of a single element is $r = e^{-(0.48\times10^{-6})(50,000)} = 0.9762858$.

Note that signal 6 is specified as the subsystem output by the last card in the cperator input. The other five signals will be removed from the probability space as soon

as practicable in the execution of the computer program. If it were desired to provide visibility for any of them they could be explicitly retained by specifying them as final signals.

Only two time points were defined for this problem, 0 and 1. Signal presence at time point 0 represents an operable system and signal presence at time point 1 indicates that the signal never arrived, i.e., failed to arrive, meaning that the system is not functional. Thus at this 50,000 hr point an output of 6_0 (signal 6 at time point 0) is a success and 6_1 is a failure.

The GO3 final distribution for this example is shown below:

TABLE 11.1.2. GO RESULTS FOR EXAMPLE 1.

FINAL EVENT TABLE (INFINITY = 1)

	SIGNALS AND THEIR VALUES
PROBABILITY	6
.0000523952	1
.9999476048	0

This example is the same one discussed in Chapter 3 in which an exact equation was derived for $P(R_1)$, the reliability of subsystem #1 - i.e.,

$$P(R_1) = 3e^{-4\lambda t} 8e^{-3\lambda t}+6e^{-2\lambda t}$$

with $\lambda t = 2.4\times10^{-2}$ at t=50,000 hrs,

$$P(R_1) = 3e^{-9.6\times10^{-2}}-8e^{-7.2\times10^{-2}}+6e^{-4.8\times10^{-2}}$$

$$= 3(0.9084640)-8(0.93053087)+6(0.95313378)$$

$$= 0.99994760.$$

This is the result achieved using the GO procedure as noted above.

11.2 Example 2 - Dual Bus Power Distribution System [9]

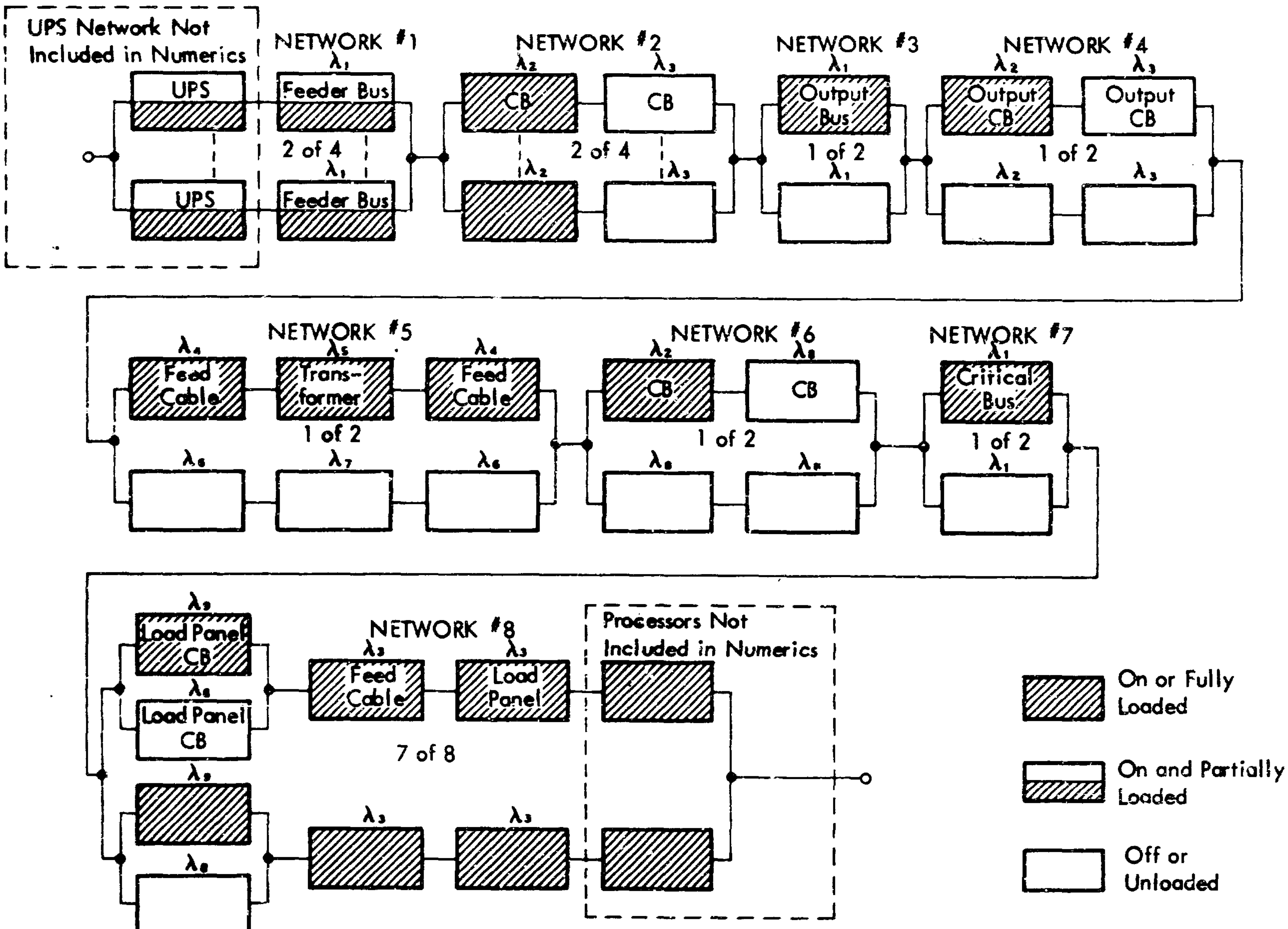
The purpose of this example is to demonstrate the use of the GO methodology to determine system MTBF.

It is desired to determine the probability that the dual bus power distribution system of Figure 11.2.1 is operational over the time interval from zero to 10^6 hours and to determine the system MTBF assuming no repairs. The system consists of eight serial networks each of which must function for proper system operation. Note that network #1 was used as Example 1.

Each element in the reliability block diagram of Figure 11.2.1 has an exponential failure distribution with failure rate λ_i, $i=1,2,...,9$ identified on the diagram. The numerical values for the λ_i's are documented in Table 11.2.1.

TABLE 11.2.1. FAILURE RATE VALUES FOR DUAL BUS POWER DISTRIBUTION SYSTEM

$\lambda_1 = 0.48\times10^{-6}$
$\lambda_2 = 1.50\times10^{-6}$
$\lambda_3 = 0.50\times10^{-6}$
$\lambda_4 = 0.13\times10^{-6}$
$\lambda_5 = 0.86\times10^{-6}$
$\lambda_6 = 0.03\times10^{-6}$
$\lambda_7 = 0.26\times10^{-6}$
$\lambda_8 = 0.40\times10^{-6}$
$\lambda_9 = 1.20\times10^{-6}$



NOTE: Diagram Drawn Showing Non-Load Sharing Mode of Operation

FIGURE 11.2.1. RELIABILITY BLOCK DIAGRAM - DUAL BUS POWER DISTRIBUTION SYSTEM.

The GO reliability diagram for this system is depicted in Figure 11.2.2. The serial nature of the eight networks is clearly evidenced.

The GO operator kind and parameter data for this example as it was entered into the computer program is reproduced here in its entirety (Table 11.2.2). Note that supertypes were used to model network #8. The kind probabilities for t = 10,000 hrs are recorded and these were altered for each run in which the system reliabilities at different elapsed times were calculated.

Since the objective is to determine a number of points of the system reliability function on the range $0 \leq t \leq 10^6$ a number of GO runs will be required for various t. A specific GO run determines the probabilities that the system is or is not functional after t hours have elapsed using two time points, 0 and 1. An output of signal 108 at time point 0 (108_0) is the event that the system is functional and 108_1 is the event that it is not. Table 11.2.3 documents the GO results for t=10,000 hrs.

TABLE 11.2.3. GO RESULTS - EXAMPLE 2
t = 10,000 Hours.

FINAL EVENT TABLE (INFINITY = 1)

	SIGNALS AND THEIR VALUES
PROBABILITY	108
.0031459148	1
.9968526632	0

TOTAL PROBABILITY = .9999985780
TOTAL ERROR = .0000011913

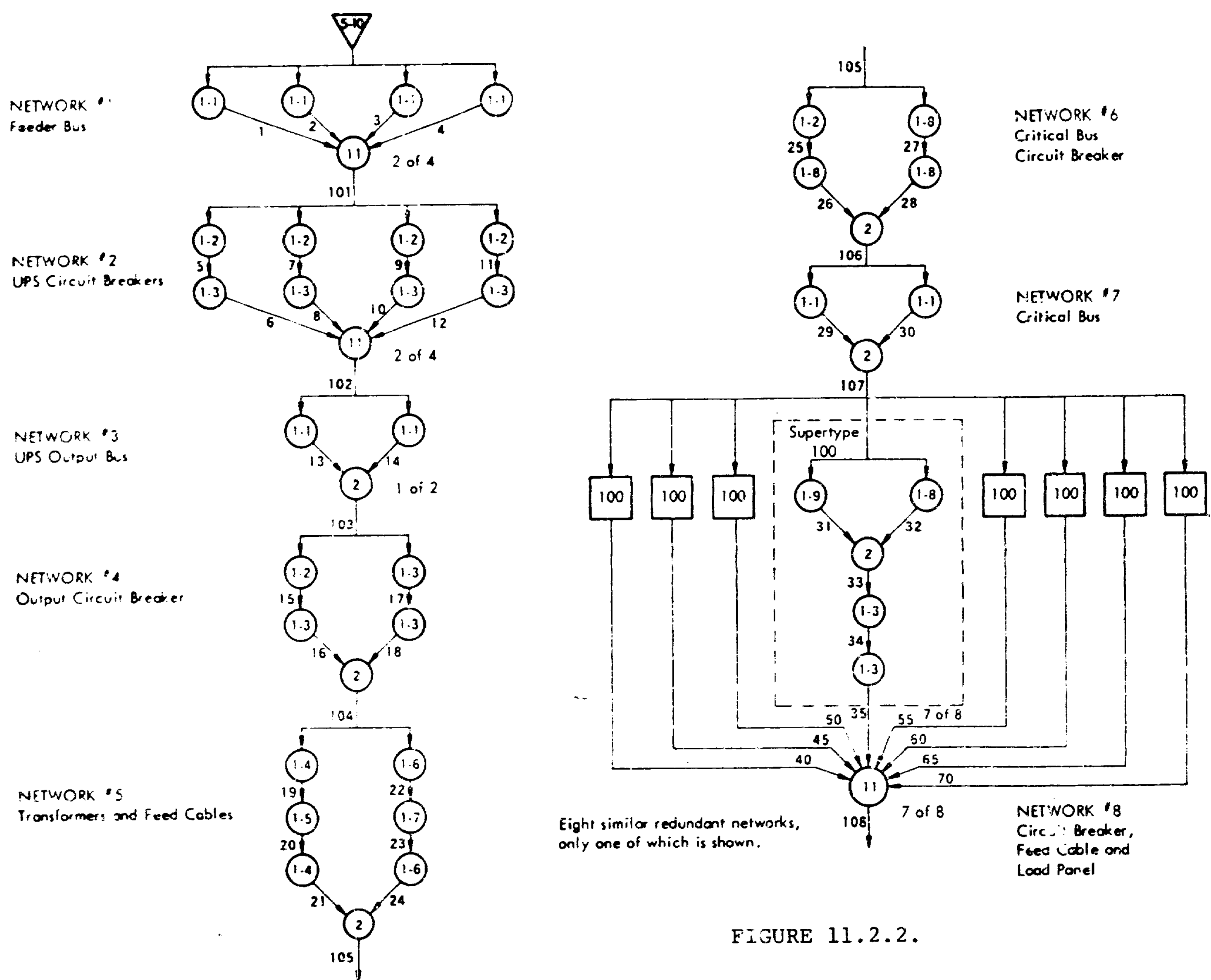


FIGURE 11.2.2.

DUAL BUS POWER DISTRIBUTION SYSTEM GO CHART

TABLE 11.2.2. GO OPERATOR, KIND & PARAMETER DATA FOR EXAMPLE 2.

```
GO APPLICATIONS EXAMPLE 2 - DUAL BUS POWER DISTRIBUTION
 $PARAM VALUES=2 $
5 10 100$
1 1 100 1$
1 1 100 2$
1 1 100 3$
1 1 100 4$
11 2 4 1 2 3 4 101$
1 2 101 5$
1 2 101 7$
1 2 101 9$
1 2 101 11$
1 3 5 6$
1 3 7 8$
1 3 9 10$
1 3 11 12$
11 2 4 6 8 10 12 102$
1 1 102 13$
1 1 102 14$
2 0 2 13 14 103$
1 2 103 15$
1 3 103 17$
1 3 15 16$
1 3 17 18$
2 0 2 16 18 104$
1 4 104 19$
1 6 104 22$
1 5 19 20$
1 7 22 23$
1 4 20 21$
1 6 23 24$
2 0 2 21 24 105$
1 2 105 25$
1 8 105 27$
1 8 25 26$
1 8 27 28$
2 0 2 26 28 106$
1 1 106 29$
1 1 106 30$
2 0 2 29 30 107$
100 -1 101 201$
1 9 101 31$
1 8 101 32$
2 0 2 31 32 33$
1 3 33 34$
1 3 34 201$
EOR
```

TABLE 11.2.2. (Continued)

```
100 0 107 35$
100 0 107 40$
100 0 107 45$
100 0 107 50$
100 0 107 55$
100 0 107 60$
100 0 107 65$
100 0 107 70$
11 7 8 35 40 45 50 55 60 65 70 108$
0 108$
EOR
GO APPLICATIONS EXAMPLE 2 - DUAL BUS POWER DISTRIBUTION
1,1,0.995212,0.004788$
2,1,0.985112,0.014888$
3,1,0.995012,0.004988$
4,1,0.998701,0.001299$
5,1,0.991437,0.008563$
6,1,0.999700,0.000300$
7,1.0.997403,0.002597$
8,1,0.996008,0.003992$
9,1,0.988072,0.011928$
10,5,1,0,1.$
EOR
GO APPLICATIONS EXAMPLE 2 - DUAL BUS POWER DISTRIBUTION
 $PARAM PMIN=1.E-8 $
EOR
```

Table 11.2.4 documents the results from these runs showing the system reliability (108_0) as a function of time. This data is plotted in Figure 11.2.3 from which one can determine the reliability at any point in time.

To determine the system MTBF certain fundamental relationships must be recalled. If $f(t)$ represents the system failure density function, then $F(t) = \int_0^t f(t)dt$ is the cumulative failure distribution. The reliability function $R(t)$ is defined as $R(t) = 1 - F(t)$, and the relationship

$$MTBF = \int_0^\infty R(t)dt$$

is generally valid. [10] Hence, knowing the failure distributions for the constituent elements of a system, if the reliability function can be determined and integrated the MTBF can be determined.

As noted the reliability function for this system was tabulated in Table 11.2.4 and plotted in Figure 11.2.3. The integral of this function can be readily obtained by numerical integration. For this example the Simpson 1/3 quadrature procedure was used [11], i.e.,

$$\int_a^b R(t)dt \approx \frac{h}{3}\ (R_0 + 4R_1 + 2R_2 + 4R_3 + 3R_4 + \ldots + 4R_{N-1} + R_N)$$

where h is the equal spaced increment in the independent variable and R_i are the tabulated functional values.

TIME (10^3 Hrs)	R	TIME (10^3 Hrs)	R
0	1.000000	330	.118777
10	.996853	340	.106706
20	.987768	350	.095705
30	.973292	360	.085701
40	.953985	370	.076624
50	.930419	380	.068405
60	.930419	390	.060979
70	.872784	400	.054281
80	.839827	410	.048252
90	.804822	420	.042836
100	.768269	430	.037977
110	.730638	440	.033626
120	.692362	450	.029737
130	.653835	460	.026266
140	.615410	470	.023172
150	.577403	480	.020420
160	.540085	490	.017974
170	.503690	500	.015804
160	.468413	510	.013881
190	.434413	520	.012179
200	.401815	530	.010675
210	.370715	540	.009348
220	.341178	550	.008177
230	.313246	560	.007147
240	.286938	570	.006240
250	.262253	580	.005444
260	.239174	590	.004745
270	.217670	600	.004132
280	.197698	700	.000993
290	.179207	800	.000223
300	.179207	900	.000047
310	.162136	1000	.000010
320	.131991		

TABLE 11.2.4. DUAL BUS POWER SYSTEM RELIABILITY CALCULATED FROM GO MODEL.

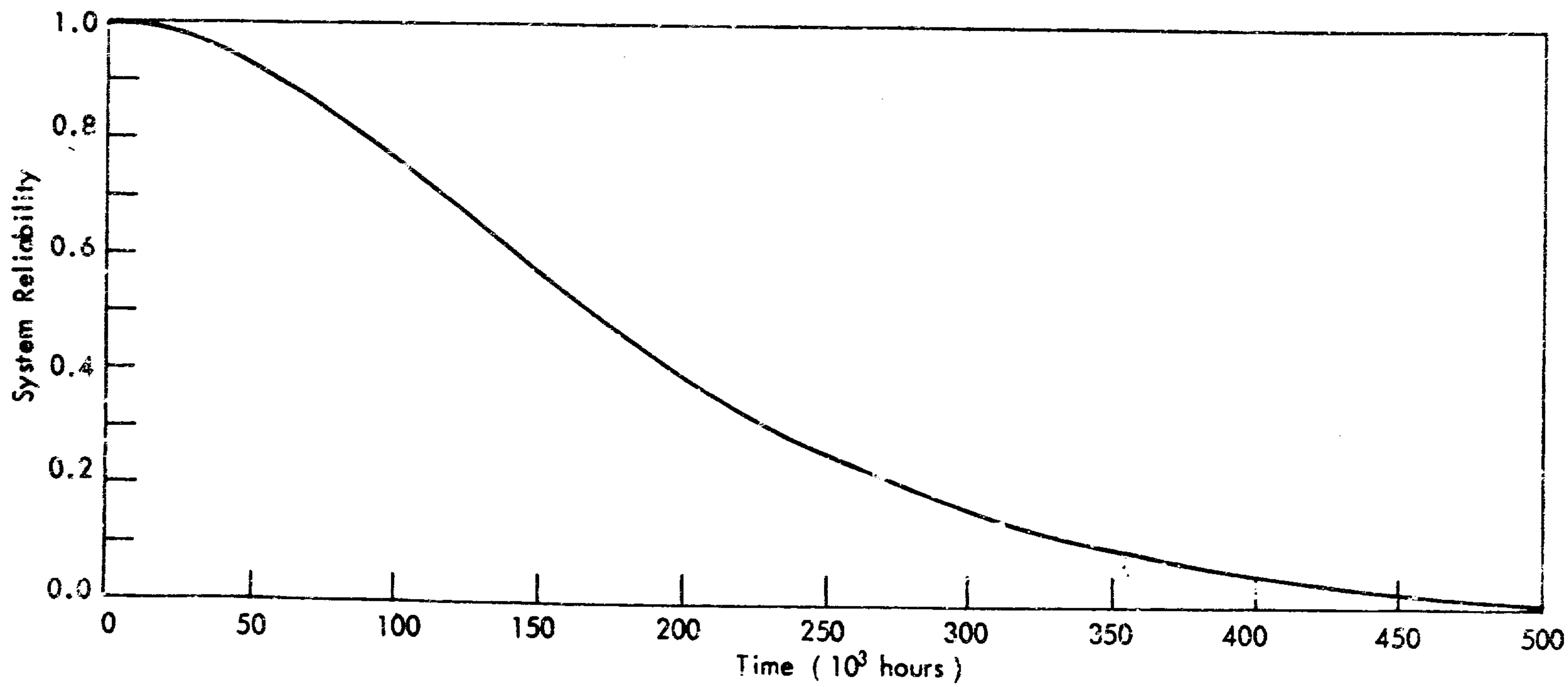


FIGURE 11.2.3. POWER DISTRIBUTION SYSTEM RELIABILITY.

This was done and the system MTBF was calculated to be 1.9041×10^5 hours.

To rigorously establish this value, the actual equation of the system in factored form was developed by writing the individual network equations as follows:

$$R_1 = 6e^{-2\lambda_1 t} - 8e^{-3\lambda_1 t} + 3e^{-4\lambda_1 t},$$

$$R_2 = 6e^{-2(\lambda_2+\lambda_3)t} - 8e^{-3(\lambda_2+\lambda_3)t} + 3e^{-4(\lambda_2+\lambda_3)t},$$

$$R_3 = 2e^{-\lambda_1 t} - e^{-2\lambda_1 t},$$

$$R_4 = e^{-(\lambda_2+\lambda_3)t} + e^{-2\lambda_3 t} - e^{-(\lambda_2+3\lambda_3)t},$$

$$R_5 = e^{-(2\lambda_4+\lambda_5)t} + e^{-(2\lambda_6+\lambda_7)t} - e^{-2(2\lambda_4+\lambda_5+\lambda_6)t},$$

$$R_6 = e^{-(\lambda_2+\lambda_8)t} + e^{-2\lambda_8 t} - e^{-(\lambda_2+3\lambda_8)t},$$

$$R_7 = 2e^{-\lambda_1 t} - e^{-2\lambda_1 t},$$

$$R_8 = 8P^7(1-P) + P^8 = 8P^7 - 7P^8,$$

$$P = e^{-\lambda_9 t} + e^{-\lambda_8 t} - e^{-(\lambda_8+\lambda_9)t}\, e^{-2\lambda_3 t},$$

$$= e^{-2(\lambda_3+\lambda_9)t} + e^{-(2\lambda_3+\lambda_8)t} - e^{-(2\lambda_3+\lambda_8+\lambda_9)t},$$

$$R_8 = 8\left(e^{-(2\lambda_3+\lambda_9)t} + e^{-(2\lambda_3+\lambda_8)t} - e^{-(2\lambda_3+\lambda_8+\lambda_9)t}\right)^7 - 7\left(e^{-(2\lambda_3+\lambda_9)t} + e^{-(2\lambda_3+\lambda_8)t} - e^{-(2\lambda_3+\lambda_8+\lambda_9)t}\right)^8$$

Thus the system reliability is

$$R = \prod_{i=1}^{8} R_i .$$

In factored form it is

$$R = \left(6e^{-2\lambda_1 t} - 8e^{-3\lambda_1 t} + 3e^{-4\lambda_1 t}\right)\left(6e^{-2(\lambda_2+\lambda_3)t} - 8e^{-3(\lambda_2+\lambda_3)t}\right.$$
$$\left. + 3e^{-4(\lambda_2+\lambda_3)t}\right)\left(2e^{-\lambda_1 t} - e^{-2\lambda_1 t}\right)^2$$
$$\left(e^{-(\lambda_2+\lambda_3)t} + e^{-2\lambda_3 t} - e^{-(\lambda_2+3\lambda_8)t}\right)$$
$$\left(e^{-(\lambda_2+\lambda_8)t} + e^{-2\lambda_8 t} - e^{-(\lambda_2+3\lambda_8)t}\right)$$
$$\left[8\left(e^{-(2\lambda_3+\lambda_9)t} + e^{-2(2\lambda_3+\lambda_8)t} - e^{-(2\lambda_3+\lambda_8+\lambda_9)t}\right)^7\right.$$
$$\left. -7\left(e^{-(2\lambda_3+\lambda_9)t} + e^{-(2\lambda_3+\lambda_8)t} - e^{-(2\lambda_3+\lambda_8+\lambda_9)t}\right)^8\right] .$$

To actually perform the integration of the 1558 explicit terms as a function of the λ_i values would be a tedious task. Instead, we again reverted to a numerical procedure. Using the system equation defined above, 101 values were calculated for R over the time interval from 0 to 10^6 hours. These were identical to those calculated from the GO model documented in Table 8.2.4 except that 36 additional points were calculated.

When the integration was performed the MTBF was calculated to be 1.9070×10^5. Hence, the previous result obtained by employing the GO methodology is established. The slight variation in the two results could have been obviated by calculating and integrating over the same number of points.

It may be of interest to note that the development of the system equation required significantly more analyst labor than that expended in creating the GO Chart and calculating the system reliability estimates. The fact that the MTBF calculated in Reference [8] for this system using the equation writing technique was incorrect may reflect the complexity and tedium required to correctly employ that procedure.

These first two examples provide visibility about how the GO procedure can be utilized to perform typical system analyses. Introducing the equations along with the GO analysis clarifies the use of the GO procedure and gives confidence that it produces the correct results.

The application here to determine system MTBF may also be of continuing interest. One is cautioned to keep in mind the restrictions for this application which are:

(1) The failure distributions with time for the components must be known or assumed;

(2) no repairs are effected;

(3) multiple runs are required.

11.3 Example 3 - Switching Network [12]

The purpose of this example is to demonstrate the capability and flexibility to handle sequenced events.

The reliability of the dual channel cross-connected switching circuit of Figure 11.3.1 will be analyzed in this section. It was originally treated in Reference 13 and subsequently addressed in Reference 12. This is a one-shot system built and installed or stored until its intended one-time use. It is desired to ascertain the probability that it will successfully function when energized given that the probabilities of component responses and arrivals of external signals are known.

The intended operation of the switching network is as follows:

1. Batteries U_A and U_B are activated to create a voltage differential on each channel.
2. External input signal W energizes actuators S_{1A} and S_{1B} which close normally open switch contacts in the six output signal circuits C_A, C_B, D_A, D_B, E_A, E_B.
3. External input signals X_A and X_B close their associated electrical contacts applying power to points A_1 and B_1 and producing outputs C_A and C_B. Power is applied to S_{2A} and S_{2B} which operate, close their respective contacts and apply power to points A_2 and B_2.
4. External input signal Y energizes actuators S_{3A} and S_{3B} enabling outputs D_A and D_B and sending power to the diodes V_A and V_B.
5. External input signal Z closes the circuit between the diodes and point A_3 enabling outputs E_A and E_B.

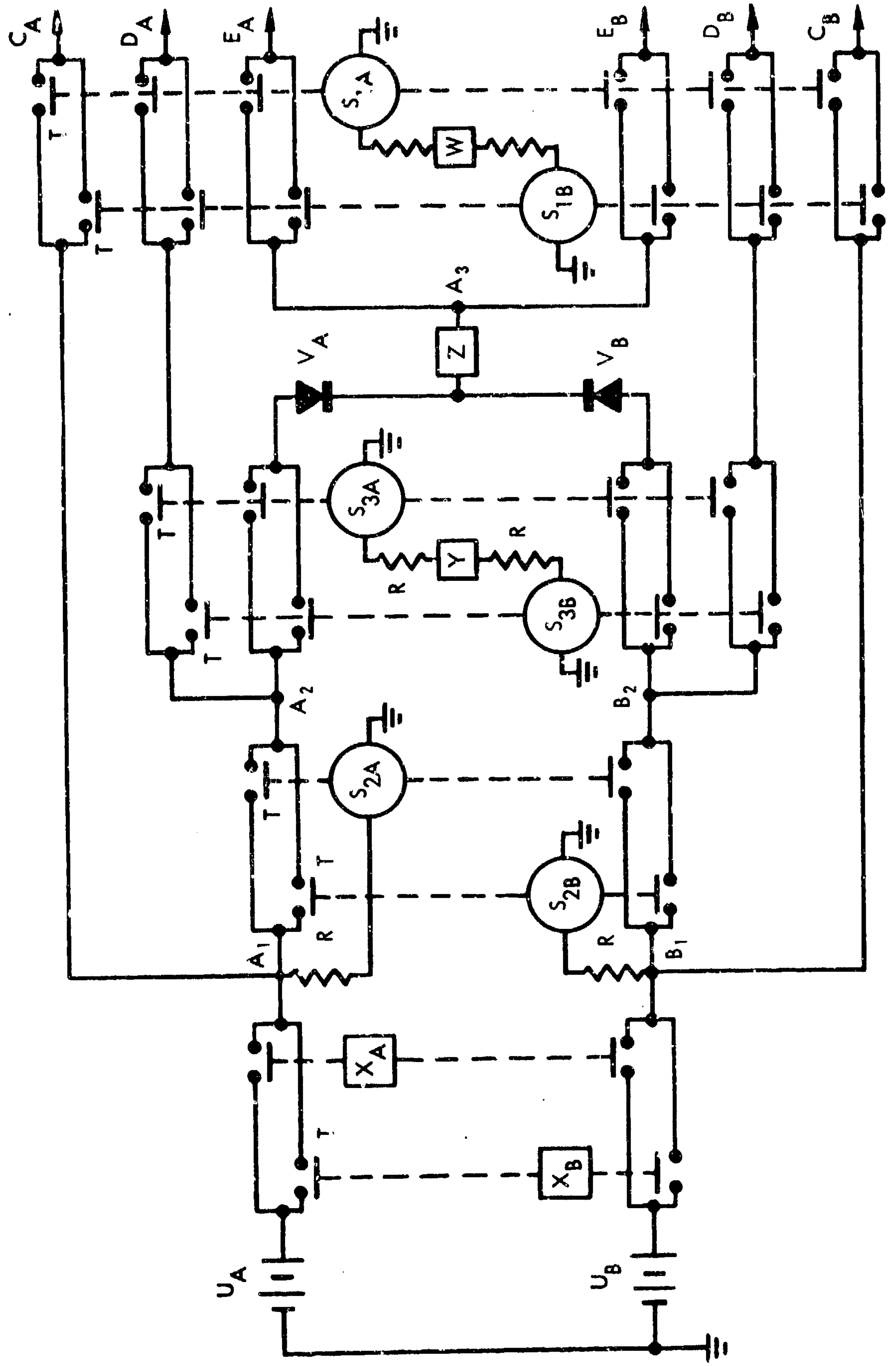


FIGURE 11.3.1. SWITCHING NETWORK.

The redundancies in the design indicate the objective to insure high reliability. The serial switches dependent upon sequenced external signals reveal that the output signals must likewise be sequenced correctly.

Successful operation of the system is defined to be output C_A or C_B (or both) and output D_A or D_B (or both) followed at least one time point later by output E_A or E_B or both.

Figure 11.3.2 is the GO Reliability Diagram of the Switching Network. The triangles identify the external signals and the components are modeled using the standard GO operators (types). The relationships of GO symbols to system components is very nearly one-to-one with the exception of some additional GO logic symbols required to combine the output signals to generate the defined successful response modes.

Note the use of the supertype 100 to model the recurrent combination of paralleled normally open switch contacts. The use of a type 9 operator to accept combinations of signals 44 and 42 where signal 44 arrives at least one time point after signal 42 may be of interest. The final signal 45, represents the system response. The time response distribution of signal 45 will be calculated by the GO program and separated into premature, success and dud system events.

The following arbitrary time reference scale is superimposed upon the system and will enable discussion of its sequential operation:

Time Point	Definition
0	Prior to initiation of the batteries and arrival of signal W.
1	Time instant at which signal W should normally arrive.

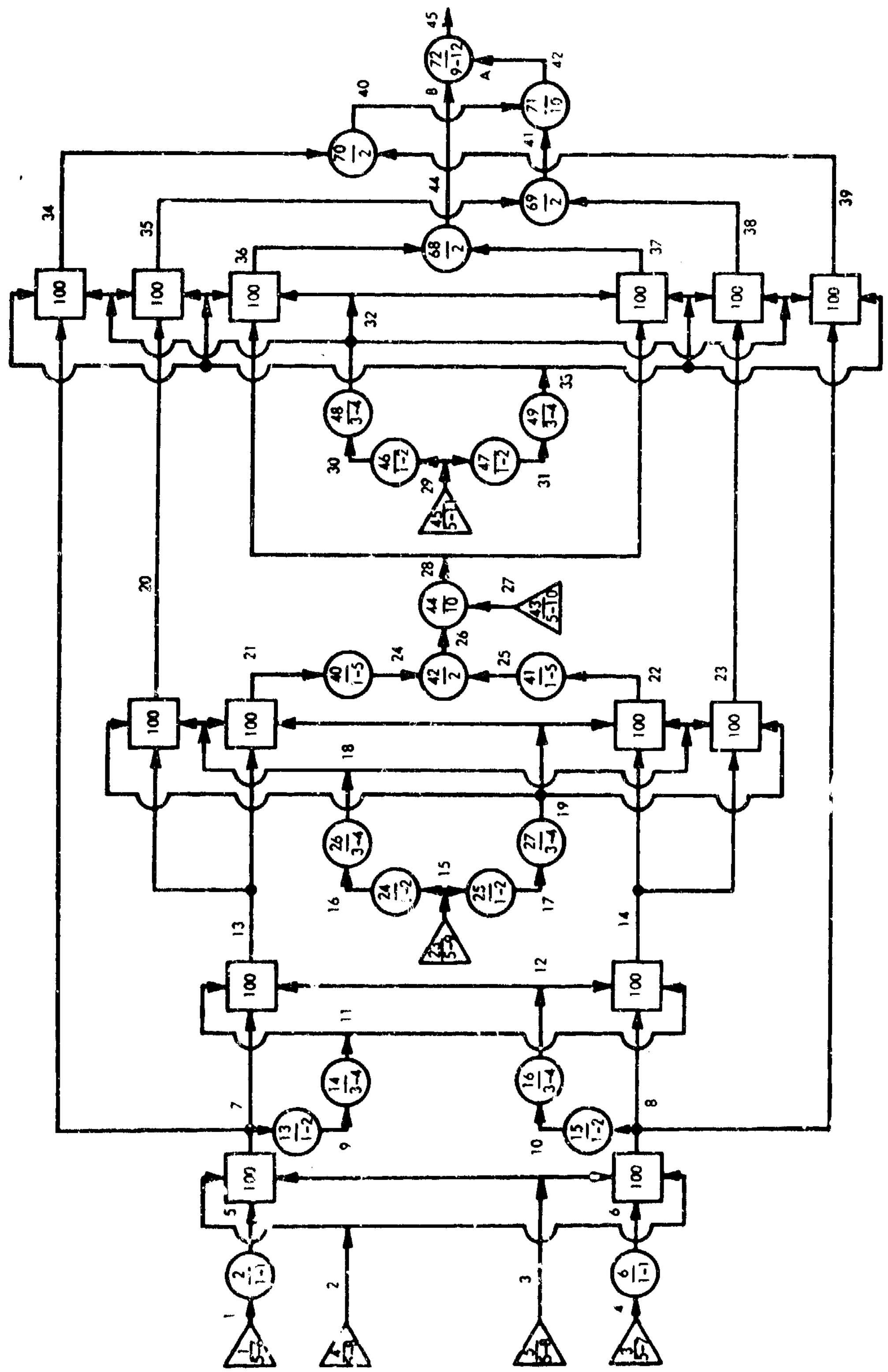


FIGURE 11.3.2. GO CHART OF SWITCHING NETWORK.

Time Point	Definition
2	Time instant between signals W and X_A, and X_B.
3	Time instant at which signals X_A and X_B should normally arrive.
4	Time instant between arrival of signals X_A, X_B, and Y.
5	Time instant at which signal Y should normally arrive.
6	Time instant between arrival of signals Y and Z.
7	Time instant at which signal Z should normally arrive.
8	Final time point of interest.
15	Never (i.e., at infinite time).

Note that the sequence of external signal arrival is expected to be W, X_A and X_B, Y, and finally Z.

It is important to stress that these defined time points are not equally spaced in time. They could be, but it is hardly ever convenient or efficient to impose equi-spaced time points. In this case, the actual times might be as follows:

Time Line (Seconds) 0 5 10 15 20 25 30 35 40 45 50

0 1 2 3 4 5 6 7 8 15
Time Points

Table 11.3.1 contains the assumed probabilities that external signals - battery initiate, W, X_A, X_B, Y and Z - arrive at each of the specified time points. The uncertainty in arrival could occur for many reasons - improper operator action, spurious electromagnetic radiation, etc.

The probabilities listed reflect the expected operational sequence. Since external signals X_A and X_B are assumed to be identical only one kind has been specified. The same is true of the battery initiating signals although they were each given a unique kind number but identical probabilities of operation.

Table 11.3.2 lists the probabilities of premature, normal and dud operation of each component. The GO type-kind symbology is specified in both tables to correlate the external signals and components on the GO reliability diagram with the physical system.

In this example all components of the same type (e.g., all switch contacts or actuators) have the same probabilistic characteristics although they are statistically independent. For example, all of the switch contacts have the same probability of operating prematurely, but if one such contact exhibits premature operation it has no effect on the probability that others also operate prematurely.

The actual card images of the GO input data for this example are listed in Table 11.3.3. Both the operator data and the kind data are documented.

TABLE 11.3.1. PROBABILITIES OF EXTERNAL SIGNALS ARRIVING AT VARIOUS TIME POINTS.

GO TYPE-KIND SYMBOLOGY		TIME POINT									
		0	1	2	3	4	5	6	7	8	15
5-11	Probability of Signal W Ocurring at Time Point i (w_i):	.0005	.9990	0.0	0.0	0.0	0.0	0.0	0.0	.0005	0.0
5-8	Probability of Signals X_A, X_B, Occurring at Time Point i (x_i):	.0005	0.0	.0010	.9900	.0005	0.0	0.0	0.0	.0080	0.0
5-9	Probability of Signal Y Ocurring at Time Point i (y_i):	.0002	.0010	.0005	.0001	.0010	.9960	.0001	.0001	.0010	0.0
5-10	Probability of Signal Z Ocurring at Time Point i (z_i):	0.0	.0002	.0001	0.0	.0001	.0005	.0001	.9980	.0010	0.0
5-6	Probability of Signal Energizing Battery U_A Occurring at Time Point i :	0.0	1.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5-7	Probability of Signal Energizing Battery U_B Occurring at Time Point i:	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9-12*	Function Operator Requiring Proper Signal Sequencing	15	1	2	3	4	5	6	15	15	15

* For the type 9 operator the column headings specify the value of $VS_2 - VS_1$ and the entries the amount by which VS_1 is to be incremented, i.e., $f(VS_2 - VS_1)$.

TABLE 11.3.2. COMPONENT RELIABILITY DATA.

GO TYPE-KIND SYMBOLOGY	DEFINITION	PROBABILITY
1-2	Probability of resistor's resistance not being excessive	.9999
	Probability of resistance of resistor being excessive	.0001
1-1	Probability of normal battery function	.9950
	Probability of battery failure	.0050
1-5	Probability of normal diode operation	.9990
	Probability of open diode	.0010
3-4	Probability of switch actuator operating in the absence of a signal	.0020
	Probability of normal switch actuator performance	.9950
	Probability of switch actuator failing to actuate when signal is applied	.0030
6-3	Probability of premature switch contact closure	.0001
	Probability of normal switch contact performance	.9998
	Probability of switch contact remaining open, even when actuated	.0001

TABLE 11.3.3. GO OPERATOR, KIND & PARAMETER DATA FOR EXAMPLE 3.

```
GO APPLICATIONS EXAMPLE 3 - SWITCHING NETWORK
 $PARAM VALUES=16 $
5 6 1$
1 1 1 5$
5 7 4$
5 8 2$
5 8 3$
100 -1 101 102 103 201 1001$
6 1001 101 102 50 $
6 1001 101 103 51 $
2 0 2 50 51 201$
EOR
1 1 4 6$
100 0 5 2 3 7 3$
100 0 6 2 3 8 3$
1 2 7 9$
3 4 9 11$
1 2 8 10$
3 4 10 12 $
100 0 7 11 12 13 3$
100 0 8 11 12 14 3$
5 9 15$
1 2 15 16$
1 2 15 17$
3 4 16 18$
3 4 17 19$
100 0 13 18 19 20 3$
100 0 13 18 19 21 3$
100 0 14 18 19 22 3$
100 0 14 18 19 23 3$
1 5 21 24$
1 5 22 25$
2 0 2 24 25 26$
5 10 27$
10 0 2 26 27 28$
5 11 29$
1 2 29 30$
1 2 29 31$
3 4 30 32$
3 4 31 33$
100 0 7 32 33 34 3$
100 0 20 32 33 35 3$
100 0 28 32 33 36 3$
100 0 8 32 33 39 3$
100 0 23 32 33 38 3$
100 0 28 32 33 37 3$
2 0 2 36 37 44$
2 0 2 35 38 41$
2 0 2 34 39 40$
10 0 2 40 41 42$
9 12 42 44 45$
EOR
```

TABLE 11.3.3. (Continued)

```
GO APPLICATIONS EXAMPLE 3 -  SWITCHING NETWORK
1,1,0.9950,0.0050$            BATTERY
2,1,0.9999,0.0001$            RESISTOR
3,6,0.9998,0.0001,0.0001$     NORMALLY OPEN CONTACTS
4,3,0.9950,0.0030,0.0020$     SWITCH ACTUATOR
5,1.0.999,0.001$              DIODE
6,5,1,1,1.$                   A CHANNEL BATTERY INITIATE SIGNAL
7,5,1,1,1.$                   B CHANNEL BATTERY INITIATE SIGNAL
8,5,5,0,0.0005,2,0.001,3,0.99,4,0.0005,8,0.008$ EXTERNAL SIGNAL SIGNAL XA OR XB
9,5,9,0,.0002,1,.001,2,.0005,3,.0001,4,.001,5,.996,6,.0001,7,.0001,8,.001$ SIG Y
10,5,7,1,.0002,2,.0001,4,.0001,5,.0005,6,.0001,7,.998,8,.0010$ EXTERNAL SIG Z
11,5,3,0,.0005,1,.9990,8,.0005$ EXTERNAL SIGNAL W
12,9,7,0,15,1,1,2,2,3,3,4,4,5,5,6,6$ FUNCTION OPERATOR FOR CORRECT SEQUENCING
EOR
GO APPLICATIONS EXAMPLE 3 - SWITCHING NETWORK
 $PARAM INTER=1, PMIN=1.E-08, SAVE=1$
EOR
```

With these inputs the results from exercising the GO program are the events that the system responded at the various time points previously defined. These results are documented in Table 11.3.4.

TABLE 11.3.4. GO RESULTS FOR EXAMPLE 3.

FINAL EVENT TABLE (INFINITY = 15)

PROBABILITY	SIGNALS AND THEIR VALUES 45
.0000005633	4
.0000036760	5
.0000995177	6
.0009984485	8
.0026440872	15
.9962463712	7

TOTAL PROBABILITY = .9999926684

TOTAL ERROR = .0000073361

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	45
4	.0000005633
5	.0000036760
6	.0000995177
7	.9962476231
8	.0009984485
15	.0026440872

Table 11.3.4 lists the computed probabilities of occurrence of signal 45 occurring at the various time points. These time points and associated probabilities are listed in order of increasing probability. The most likely event, 45_7, is listed last. It represents successful system operation. The probability of successful system operation, i.e., its reliability, is calculated to be 0.996246.

The probability of failure, i.e., that signal 45 never arrived is 2.6441×10^{-3} identified as event 45_{15}. Three premature events are noted (45_4, 45_5, and 45_6) and one late event, 45_8. With reference to the arbitrary chronology utilized all events can be interpreted as prematures, successes, or duds. In this case, the prematures can be further differentiated if desired.

In this example, events with probabilities of occurrence greater than 1×10^{-8} have been retained. Others have been discarded and the resultant calculated values are approximations to the actual value which would result without truncation. A bound on the magnitude of the error introduced is obtained by summing the probabilities of occurrence of the events listed and subtracting from unity. In this example, the total error is 7.34×10^{-6}.

This total error does not appear at only one time instant but is distributed over them in a manner which can only be determined by avoiding such truncation. The maximum error in the probability of occurrence of any event is less than the total.

To clarify the effect of changing the threshold of retention, the program adjusted parameter PMIN was set at 1×10^{-10} and the problem was rerun. The results of Table 11.3.5 document the effect of retaining additional terms formerly discarded. The maximum error is now 2.65×10^{-6}. The first run at PMIN = 1.0×10^{-8} required 7.8 seconds of central processor time for program GO3 execution. The second required 16.4 seconds.

TABLE 11.3.5. EXAMPLE 3 RESULTS WITH PMIN = 1×10^{-10}.

FINAL EVENT TABLE (INFINITY = 15)

SIGNALS AND THEIR VALUES

PROBABILITY	45
.0000000007	2
.0000005767	4
.0000037400	5
.0000998040	6
.0009987213	8
.0026467116	15
.9962501811	7

TOTAL PROBABILITY = 0.9999997354
TOTAL ERROR = 0.0000002646

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	45
2	.0000000007
4	.0000005767
5	.0000037400
6	.0000998040
7	.9962501811
8	.0009987213
15	.0026467116

11.4 Example 4 - Fault Tree Evaluation [2]

The purpose of this example is to demonstrate how fault trees are evaluated using GO.

The example used here is the sample fault tree used in Reference 2. A reproduction of this tree is shown in Figure 11.4.1.

A GO model for the fault tree (Figure 11.4.2) was created as follows:

a. Two signal values were used, 0 representing a fault and 1 a nonfault. The reason for reversing the definition of fault and nonfault is that fault combinations rather than successes are being modeled and propagated through the tree.

b. OR gates were modeled by type 2 operators. Thus, if at least one input signal is faulty (value=0), the output is faulty.

c. AND gates were modeled by type 10 operators. Thus, if at least one input signal is nonfaulty (value =1), the output is nonfaulty - that is, all inputs must be faulty in order to produce a faulty output.

d. The components (1 through 10 in Figure 11.4.1) were modeled by type 5 operators, all of the same kind. The outputs of each operator were 0 (with probability 0.1) and 1 (with probability 0.9).

The actual input data cards for this example are depicted in Table 11.4.1.

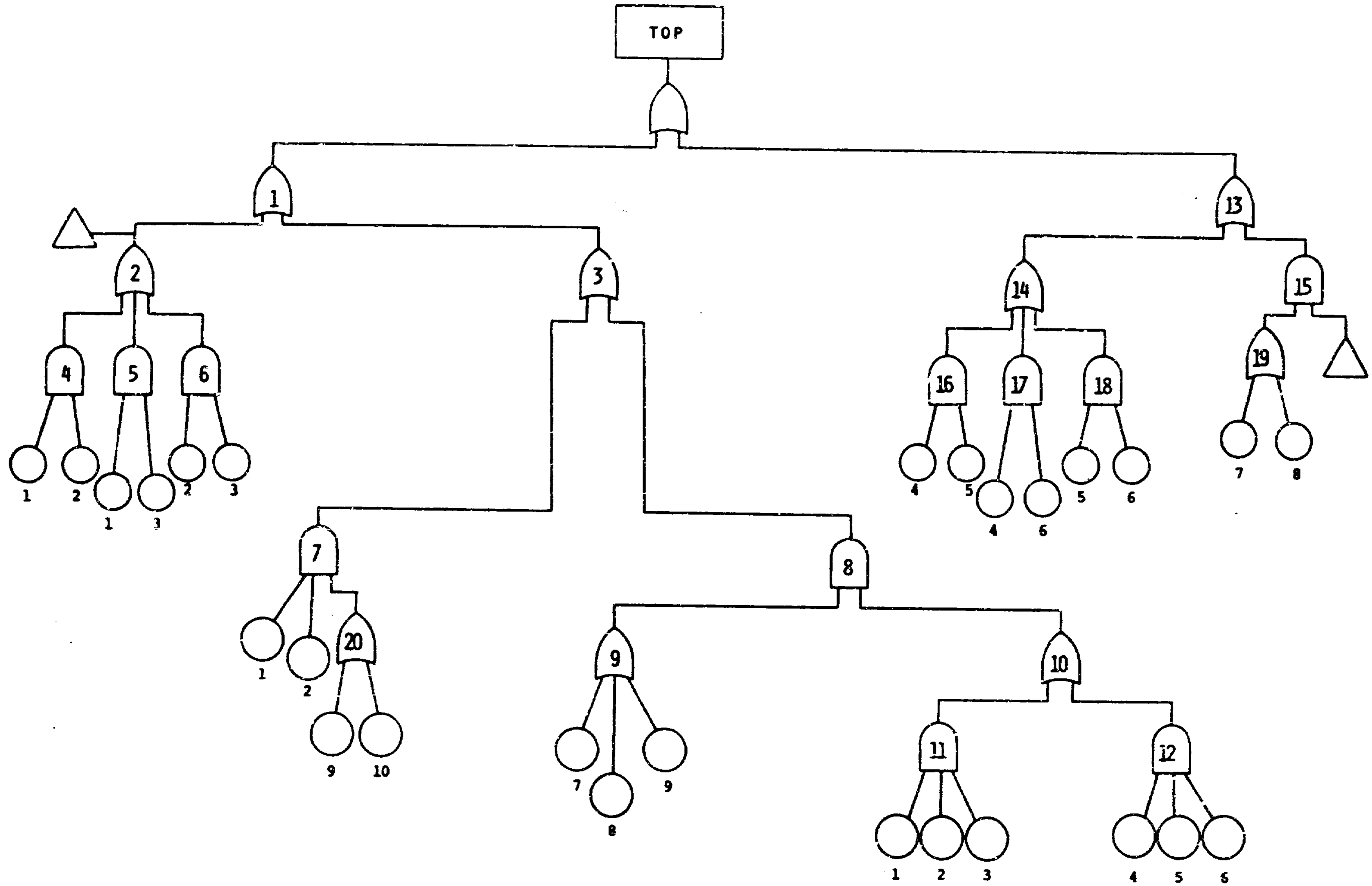


FIGURE 11.4.1. SAMPLE FAULT TREE. (From Reference 2).

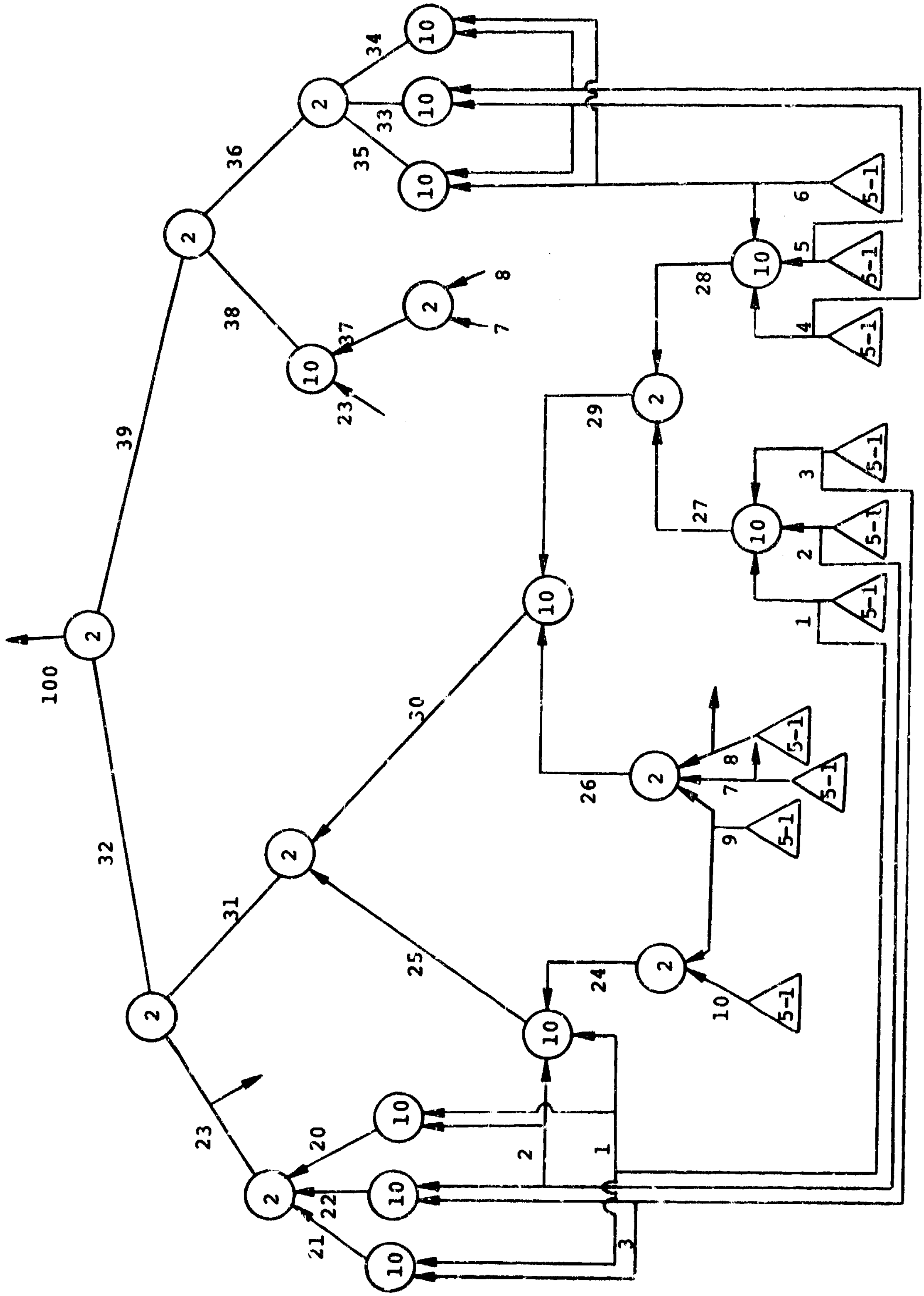


FIGURE 11.4.2. GO DIAGRAM OF SAMPLE FAULT TREE.

TABLE 11.4.1. GO INPUT DATA FOR EXAMPLE 4.

```
GO APPLICATIONS EXAMPLE 4 - FAULT TREE EVALUATION
 $PARAM VALUES=2$
5 1 1 $
5 1 2 $
5 1 3 $
10 0 2 1 2 20$
10 0 2 1 3 21$
10 0 2 2 3 22$
2 0 3 20 21 22 23$
10 0 3 1 2 3 27$
5 1 9$
5 1 10$
2 0 2 9 10 24$
10 0 3 1 2 24 25$
5 1 7$
5 1 8$
2 0 3 7 8 9 26$
2 0 2 7 8 37$
5 1 4$
5 1 5$
5 1 6$
10 0 3 4 5 6 28$
10 0 2 4 5 33$
10 0 2 4 6 34$
10 0 2 5 6 35$
2 0 3 33 34 35 36$
10 0 2 23 37 38$
2 0 2 36 38 39$
2 0 2 27 28 29$
10 0 2 26 29 30$
2 0 2 25 30 31$
2 0 2 23 31 32$
2 0 2 32 39 100$
0 100$
EOR
GO APPLICATIONS EXAMPLE 4 - FAULT TREE EVALUATION
1 5 2 0 0.1 1 0.9$
EOR
GO APPLICATIONS EXAMPLE 4 - FAULT TREE EVALUATION
 $PARAM PMIN=0.$
EOR
```

The results from the GO program are tabulated in Table 11.4.2.

TABLE 11.4.2. GO RESULTS FOR EXAMPLE 4.

FINAL EVENT TABLE (INFINITY = 1)	SIGNALS AND THEIR VALUES
PROBABILITY	100
0.0552160000	0
0.9447840000	1

The probability of a system fault or failure occurring is 0.055216. The probability of successful system operation is 0.944784.

11.5 Example 5 - Fan Control System

The purpose of this example is to demonstrate the use of the type 16 and 17 operators.

The type 16 and 17 operators provide the capability to treat what might be termed inverse operations. With all other operators the inputs and outputs are conceived as the arrival of signals, i.e., going from an off-to-on condition.

In many systems, however, events of interest model signal cessation. Typical of these are alarm, control and monitor systems which detect stoppages.

To model such systems in a straightforward manner, the type 16 and 17 operators were created as being logically consistent and complementary to other operators. The type 16 operator may be conceptualized as an actuated normally open switch (an actuated type 6). The type 17 operator may be conceptualized as an actuated normally closed switch (an actuated type 7).

In each case, the primary input is an on-to-off signal, i.e., a signal which terminates. The secondary input in each case could be either an off-to-on or an on-to-off input. The generated output for a type 16 is an on-to-off signal and that for a type 17 an off-to-on signal.

Consider the Fan Control System of Figure 11.5.1. When a temperature sensor detects excessive fan temperature the fan is shut off and a warning light is turned on.

The GO chart for this system is shown in Figure 11.5.2. The key to correct application of these operators is to keep in mind the nature of the several signals. That is, signals 1,3,4,5,6 and 7 are on-to-off signals. Their associated values reflect when they cessate. Signals 9 and 10 are off-to-on signals and signal 2 could be either.

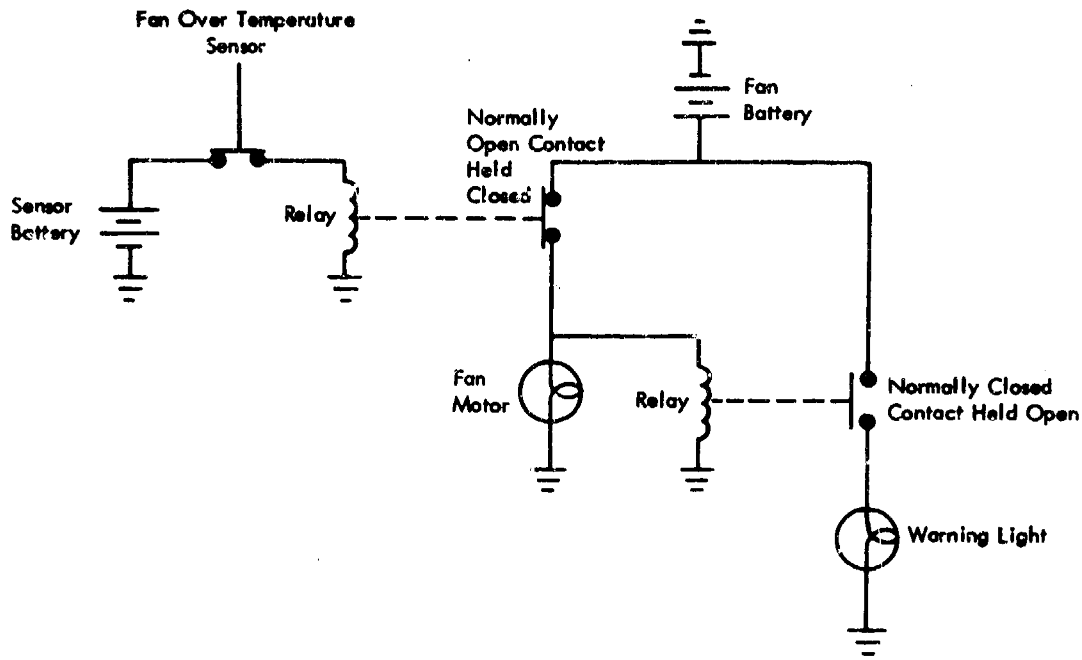


FIGURE 11.5.1. FAN CONTROL SYSTEM.

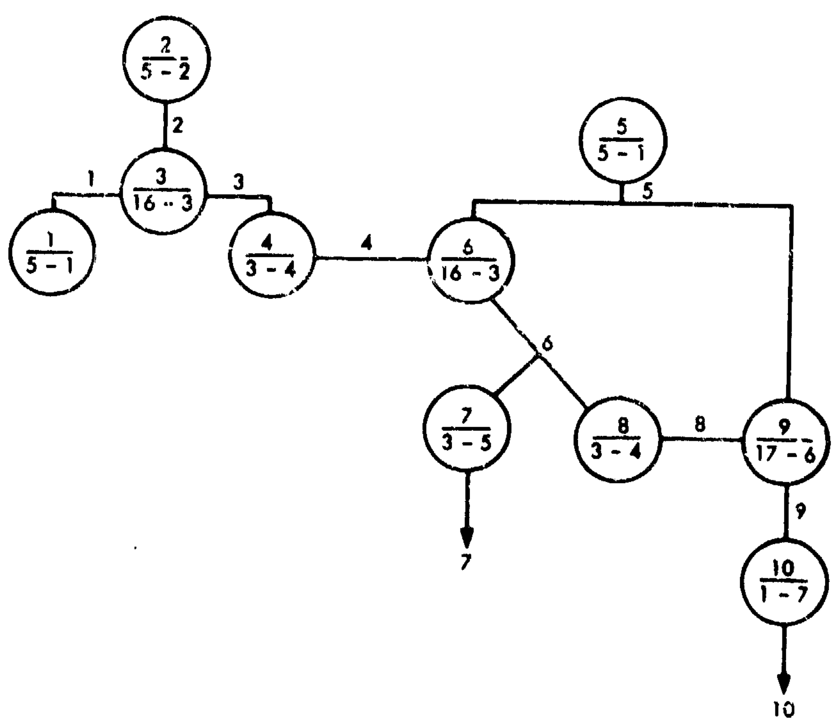


FIGURE 11.5.2. GO CHART FAN CONTROL SYSTEM.

The input data for this example is shown in Table 11.5.1. Note from the parameter card that three time points were used with infinity=2. The overheat event is prescribed to occur at time point 1 and no delays are introduced. Therefore, the fan should shut off (signal 7) and the warning light should come on (signal 10) at time=1.

The GO results are listed in Table 11.5.2. The most likely event $7_1 10_1$ indicates that with probability 0.455644 the fan turns off and the warning light comes on when the overtemperature condition is sensed. The next most likely event, $7_0 10_0$, is that both such events occur prematurely, etc.

TABLE 11.5.1. INPUT DATA FOR EXAMPLE 5.

```
GO APPLICATIONS EXAMPLE 5 - FAN CONTROL SYSTEM
 $PARAM INFIN=2  $
5 1 1 $
5 2 2 $
16 3 1 2 3 $
3 4 3 4 $
5 1 5 $
16 3 5 4 6 $
3 5 6 7 $
3 4 6 8 $
17 6 5 8 9 $
1 7 9 10 $
0 7 10 $
EOR
GO APPLICATIONS EXAMPLE 5 - FAN CONTROL SYSTEM
1 5 2 0 .05 2 .95 $ POWER SOURCES
2 5 1 1 1 $ OVERHEAT EVENT
3 16 .9 .05 .05 $ N/O ACTUATED CONTACT
4 3 .9 .03 .07 $ RELAY COIL
5 3 .9 .0 .1 $ FAN
6 17 .9 .05 .05 $ N/C ACTUATED CONTACT
7 1 .95 .05 $ WARNING LIGHT
GO APPLICATIONS EXAMPLE 5 - FAN CONTROL SYSTEM
 $PARAM  $
EOR
```

TABLE 11.5.2. GO OUTPUT FOR EXAMPLE 5

FINAL EVENT TABLE (INFINITY = 2)

	SIGNALS AND THEIR VALUES	
PROBABILITY	7	10
.0105987863	2	0
.0506271364	C	1
.0635651823	1	0
.0729208403	1	2
.0881323387	2	2
.0903544329	0	2
.1681570558	0	0
.4556442274	1	1

TOTAL PROBABILITY = 1.0000000000
TOTAL ERROR = .0000000000

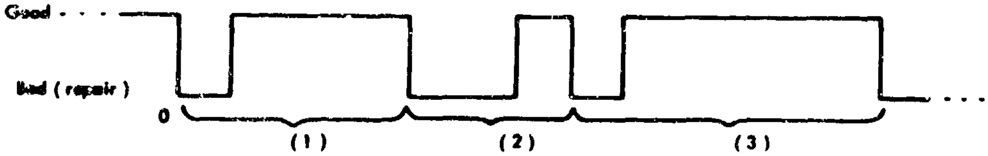
INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	7	10
0	.3091386250	.2423210244
1	.5921302500	.5062713637
2	.0987311250	.2514076119

11.6 Example 6 - Availability Analysis [15]

The purpose of this example analysis is to demonstrate how the GO methodology can be used to determine system availability where component availabilities are known. As used here, availability is the probability that a system or equipment used under stated conditions in the actual support environment will operate satisfactorily at any given time.

Consider a time history of the operation of an equipment as shown below:



Here the mean-time-between-failure (MTBF) is the average length of segments (1), (2), (3), etc., and the mean-time-to-repair* (MTTR) is the average length of "bad" segments. The availability, A, of an equipment is then

$$A = \frac{MTBF-MTTR}{MTBF}$$

It might be preferable to use Mean-Time-To-Failure (MTTF) which is the average length of "good" segments. Then

$$A = \frac{MTTF}{MTTF+MTTR}$$

*Actually, mean-down-time (MDT)

From these definitions it follows that if the MTBF or MTTF and the MTTR or MDT are known for any equipment, its availability can be expressed. For any system where the availabilities of the constituent elements are known, the GO methodology can be used to obtain the system availability.

In this case the kind probabilities will be component availabilities instead of reliabilities as in other cases. The entries then reflect that an equipment is either available or unavailable and the concept of premature has no meaning. That is, presumably both the premature and dud states of a component render it unavailable and the values failure modes.

When using the GO methodology to compute availability only two time points would be required; time point 0 to reflect availability and time point 1 to indicate that the system is unavailable.

Consider the system of Figure 11.6.1. It is a typical air system for a demonstration coal gasification plant. It is desired to ascertain the availabilities of air to the letdown hoppers and air to the coal transport and air instrumentation controls given that the availabilities of the air compressors, etc., are known.

Assume that the MTBF and MTTR for the several equipments are as documented in Table 11.6.1. From them the equipment availabilities (also shown in the table) are calculated.

The GO availability diagram for this system is shown in Figure 11.6.2. Signal 8 represents the availability of air to the letdown hoppers and signal 15 the availability of air to the coal transport and instrumentation. The GO operators model the availabilities of the several equipments.

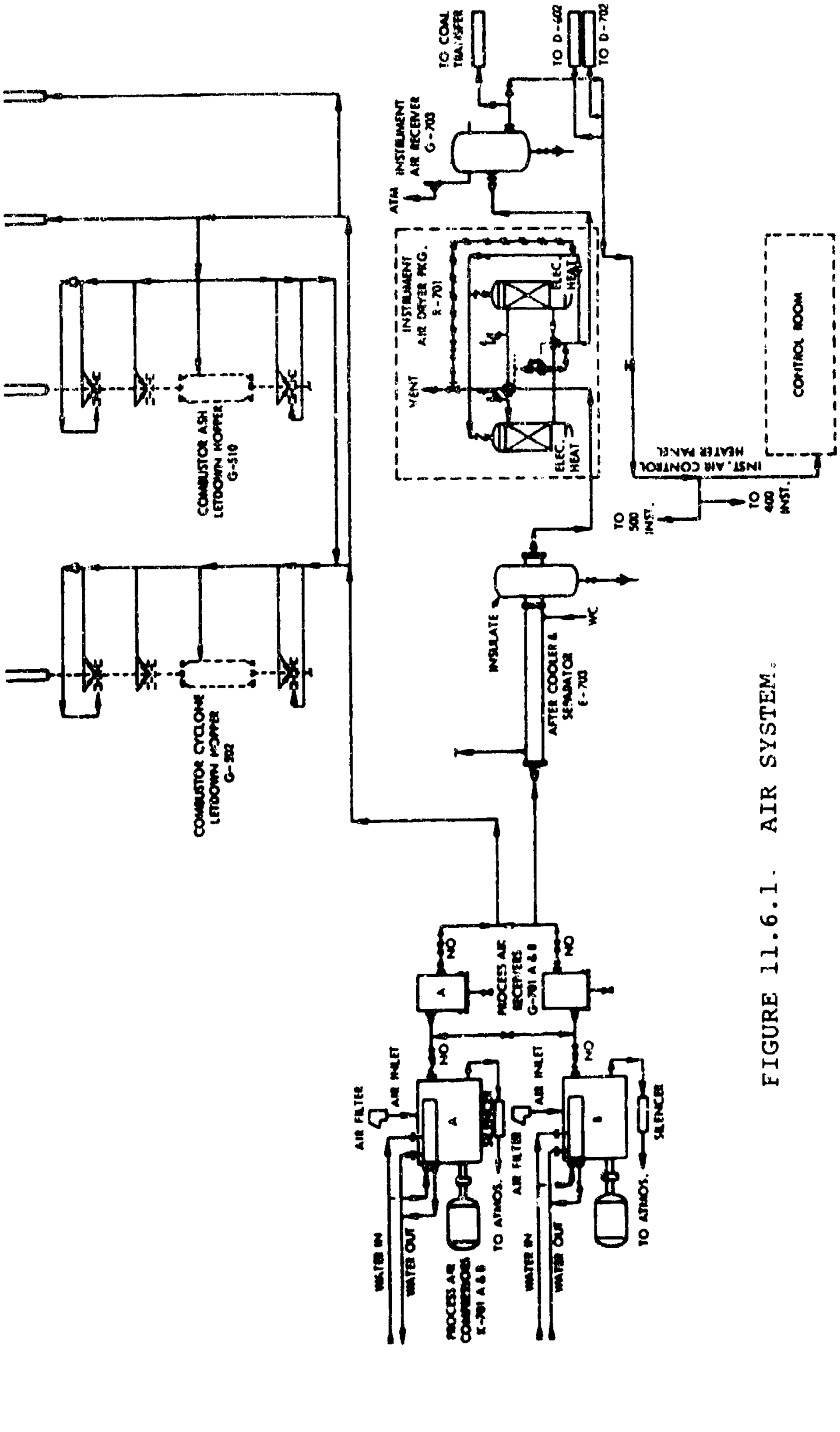


FIGURE 11.6.1. AIR SYSTEM.

TABLE 11.6.1. COMPONENT FAILURE AND REPAIR DATA FOR EXAMPLE 6.

	MTBF (hrs)	MTTR(hrs)	Availability
Air Compressor	2920	48	0.983562
Air Receiver	43800	168	0.996164
After Cooler & Separator	8760	72	0.991781
Drying Tower	4380	4	0.999087
Instrument Air Dryer	730	4	0.994521
Instrument Air Re- ceiver	43800	168	0.996164

The input operator and kind data is documented in Table 11.6.2 by portraying the actual card images. The type 5 operators represent the availabilities of air, cooling water and electricity which are assumed to be 100%.

The calculated availabilities for signals 8 and 16 are shown in Table 11.6.3.

Table 11.6.3 indicates that the probability that both air requirements are satisfied is 0.982161 at any random instant in time. Another interpretation is that both requirements are met 98.2161% of the time so the system is expected to be available 8603.7 hours in a year's time (8760 hr), etc.

The availability of air to the letdown hoppers (signal 8) is seen to be quite high, 0.999592. The availability of air to the letdown hoppers is a prerequisite for air to the coal transport and instrumentation control subsystems to be available. These system availabilities are thus completely characterized by the GO results as a function of the system configuration and the availabilities of the constituent components.

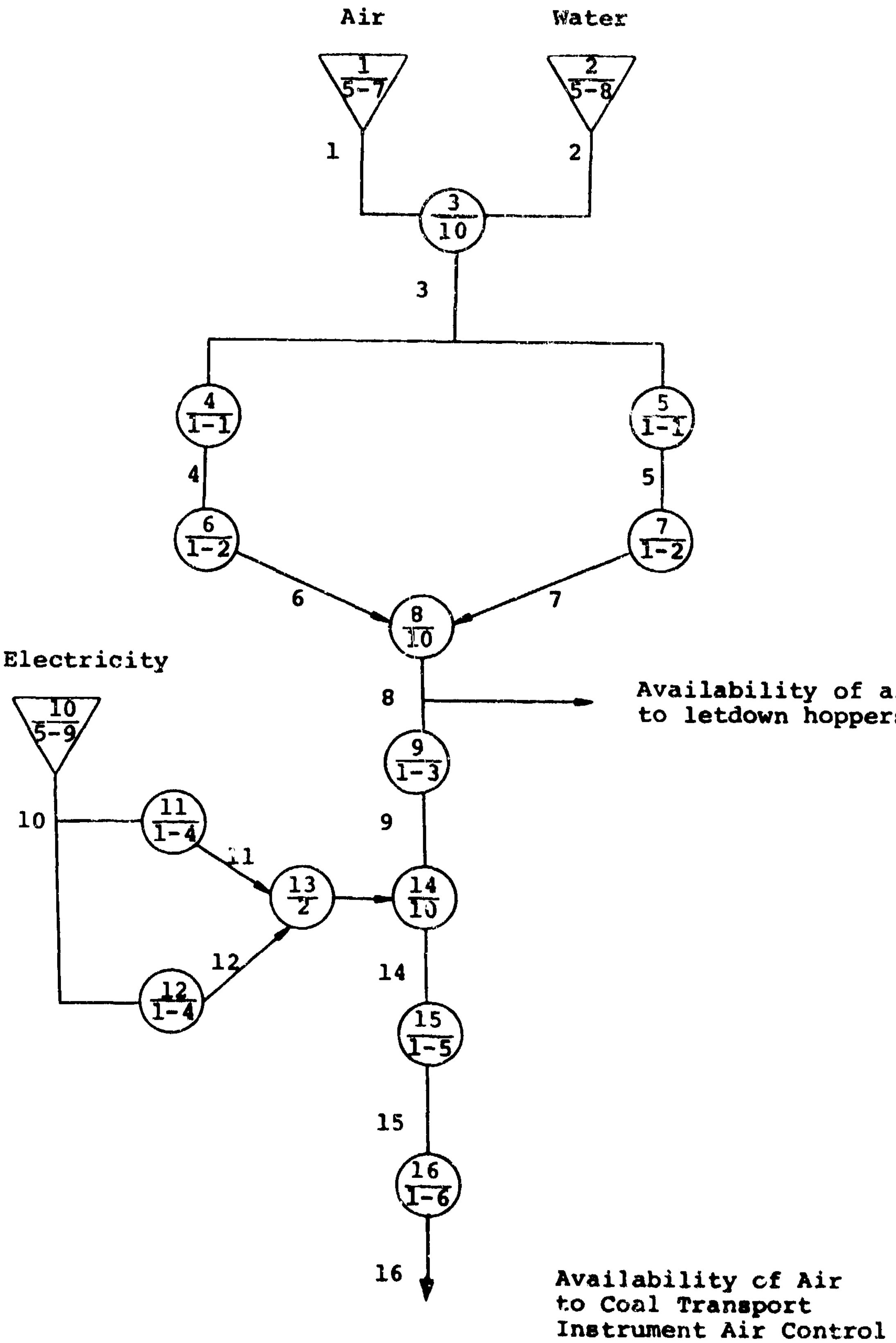


FIGURE 11.6.2. GO AVAILABILITY DIAGRAM FOR AIR SYSTEM.

TABLE 11.6.2. INPUT DATA FOR EXAMPLE 6.

```
GO APPLICATIONS EXAMPLE 6 - AIR SYSTEM AVAILABILITY
 $PARAM VALUES=2 $
5 7 1$
5 8 2$
10 0 2 1 2 3$
1 1 3 4$
1 1 3 5$
1 2 4 6$
1 2 5 7$
2 0 2 6 7 8$
1 3 8 9$
5 9 10$
1 4 10 11$
1 4 10 12$
2 0 2 11 12 13$
10 0 2 9 13 14$
1 5 14 15$
1 6 15 16$
0 8 16$
EOR
GO APPLICATIONS EXAMPLE 6 - AIR SYSTEM AVAILABILITY
1,1,0.983562,0.016438$ PROCESS AIR COMPRESSOR AVAILABILITY
2,1,0.996164,0.003836$ PROCESS AIR RECEIVER AVAILABILITY
3,1,0.991781,0.008219$ AFTER COOLER AND SEPARATOR AVAILABILITY
4,1,0.999087,0.000913$ DRYING TOWER AVAILABILITY
5,1,0.994521,0.005479$ INSTRUMENT AIR DRYER AVAILABILITY
6,1,0.996164,0.003836$ INSTRUMENT AIR RECEIVER AVAILABILITY
7,5,1,0,1.$ AIR AVAILABILITY
8,5,1,0,1.$ COOLING WATER AVAILABILITY
9,5,1,0,1.$ ELECTRICITY AVAILABILITY
EOR
GO APPLICATIONS EXAMPLE 6 - AIR SYSTEM AVAILABILITY
 $PARAM PMIN=0.0,INTER=1$
EOR
```

TABLE 11.6.3. GO RESULTS FOR EXAMPLE 6.

FINAL EVENT TABLE (INFINITY = 1

	SIGNALS AND THEIR VALUES	
PROBABILITY	8	16
.0004084823	1	1
.0174302915	0	1
.9821612263	0	0

TOTAL PROBABILITY = 1.0000000000
TOTAL ERROR = .0000000000

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	8	16
0	.9995915177	.9821612263
1	.0004084823	.0178387737

11.7 Example 7 - Standby Equipment

The purpose of this example is to demonstrate how standby equipment may be modeled.

A system with an identical redundant standby equipment is shown in Figure 11.7.1.

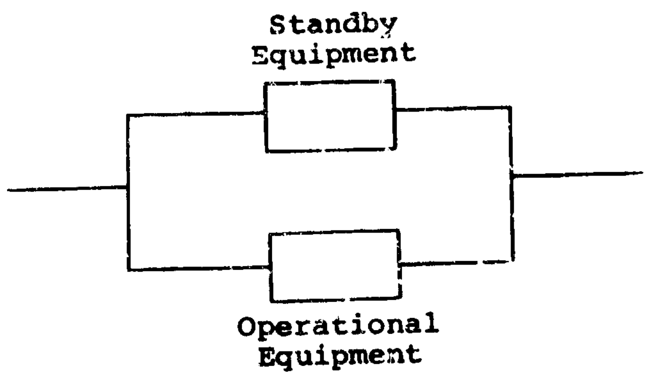


FIGURE 11.7.1. STANDBY EQUIPMENT.

A GO model of this system could be constructed as shown in Figure 11.7.2.

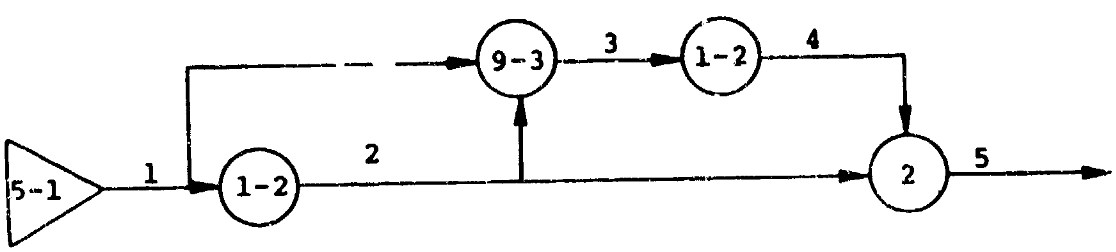


FIGURE 11.7.2. GO MODEL OF STANDBY EQUIPMENT.

The input data deck for this example is shown in Table 11.7.1.

TABLE 11.7.1. GO DATA DECK FOR EXAMPLE 7.

```
GO APPLICATIONS EXAMPLE 7 - STANDBY EQUIPMENT
 $PARAM VALUES=5$
5 1 1$
1 2 1 2$
9 3 1 2 3$
1 2 3 4$
2 0 2 2 4 5$
0 5$
EOR
GO APPLICATIONS EXAMPLE 7 - STANDBY EQUIPMENT
1 5 2 1 0.98 4 0.02$
2 1 0.95 0.05$
3 9 5 0 4 1 0 2 0 3 0 4 0$
EOR
GO APPLICATIONS EXAMPLE 7 - STANDBY EQUIPMENT
 $PARAM PMIN=1.E-10$
EOR
```

The output data for this example is shown in Table 11.7.2.

TABLE 11.7.2. FINAL EVENT TABLE FOR EXAMPLE 7.

FINAL EVENT TABLE (INFINITY=4)

	SIGNALS AND THEIR VALUES
PROBABILITY	5
0.0224500000	4
0.9775500000	1

One notes that this is precisely the value which would have been obtained if the two equipments were actively redundant. The difference is that the second equipment's usage is dependent upon failure of the first.

In actual use situations the first equipment may have been used to the time of wear out or periodic maintenance and may have a lower reliability than the standby equipment. Usually there would be some switching required to take the failed equipment off line and begin operating the alternate device with possible delay. A more complete model might be as shown in Figure 11.7.3.

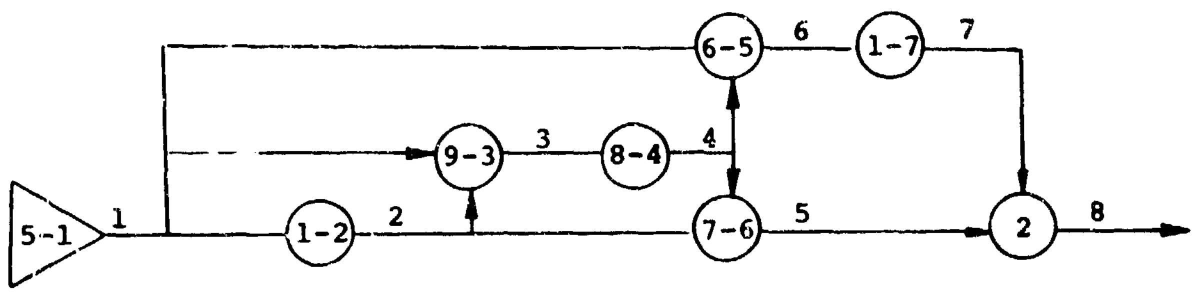


FIGURE 11.7.3. SECOND GO MODEL OF STANDBY EQUIPMENT.

Here the type 9 as before represents a sensor which detects the malfunction on the first equipment. The type 8 represents an operator response to switch out the failed equipment (1-2) and switch in the redundant equipment (1-7).

The input data deck for the second model is shown in Table 11.7.3.

The results in Table 11.7.4 show a system unreliability of 0.032091. The probability that the on-line equipment fails and the auxiliary equipment is properly switched into the system is 0.045648. The probability that the system operates successfully using either equipment is 0.967909.

TABLE 11.7.3. GO INPUT DATA FOR EXAMPLE 7(2).

```
GO APPLICATIONS EXAMPLE 7 - STANDBY EQUIPMENT(2)
 $PARAM VALUES=5$
5 1 1$
1 2 1 2$
9 3 1 2 3$
8 4 3 4$
7 6 2 4 5$
6 5 1 4 6$
1 7 6 7$
2 0 2 5 7 8$
0 8$
EOR
GO APPLICATIONS EXAMPLE 7 - STANDBY EQUIPMENT(2)
1 5 2 1 0.98 4 0.02$
2 1 0.95 0.05$
3 9 5 0 4 1 0 2 0 3 0 4 0 $
4 8 2 1 0.98 3 0.02$
5 6 0.97 0.02 0.01$
6 7 0.97 0.02 0.01$
7 1 0.98 0.02$
EOR
GO APPLICATIONS EXAMPLE 7 - STANDBY EQUIPMENT(2)
 $PARAM PMIN=1.E-10$
EOR
```

TABLE 11.7.4. GO RESULTS FOR EXAMPLE 7(2).

```
FINAL EVENT TABLE (INFINITY = 4)

                      SIGNALS AND THEIR VALUES

PROBABILITY      8

.0320907500      4
.0456478120      2
.9222614380      1
-------------------------------------------

TOTAL PROBABILITY = 1.0000000000
TOTAL ERROR =        .0000000000

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.       8

 1       .9222614380
 2       .0456478120
 4       .0320907500
```

11.8 Example 8 - Network Analysis [16]

The purpose of this example is to demonstrate how the GO program can be used to analyze networks.

Determination of the probability of successful information flow between two nodes in a complex network having nonperfect connecting links is a classical problem. Such networks may consist of computers, radios, radars, teletypes, telephones, etc., in extensive and complicated arrangements. Attempts to directly specify and partition all system states are generally frustrated because of the astronomical numbers of such combinations. In practice, even for relatively small systems, approximation techniques must be employed to render the calculations tractable.

Consider the network graph of Figure 11.8.1 from Reference 17. The nodes are designated with letters and the connecting media by line segments (branches). The branch reliabilities are specified by the numbers beside each line segment. It is assumed for this example that nodes have no unreliability, that branches are either successful or unsuccessful, that branch failures are statistically independent and that information flow is not directed.

It is desired to ascertain the probability that information sent from node s reaches node t.

To aid in translating the original network of Figure 11.8.1 into a representative GO chart. The block diagram of Figure 11.8.2 was prepared. Nodes have become components. The GO chart of Figure 11.8.3 is now prepared where signal 1 represents node s and signal 20, node t, etc. The several type 1 components represent the various network branches and have the associated branch reliabilities.

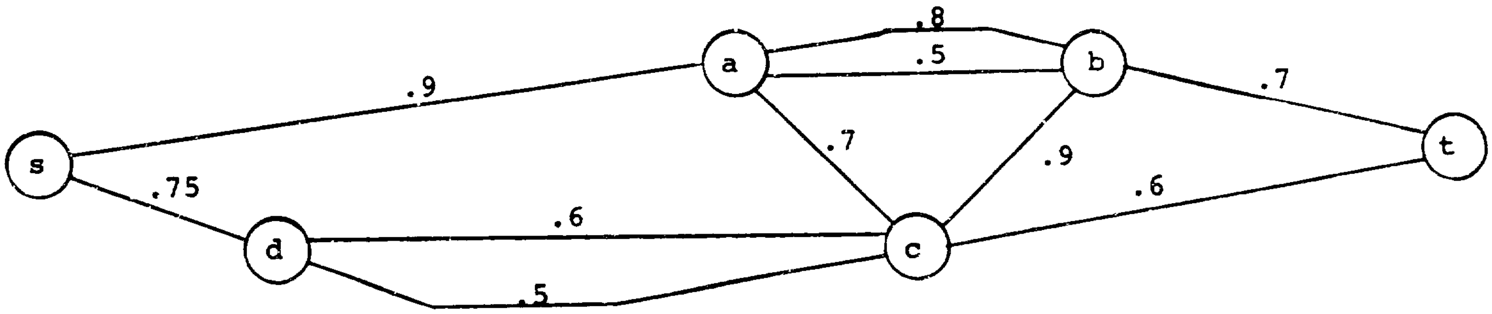


FIGURE 11.8.1. NETWORK GRAPH

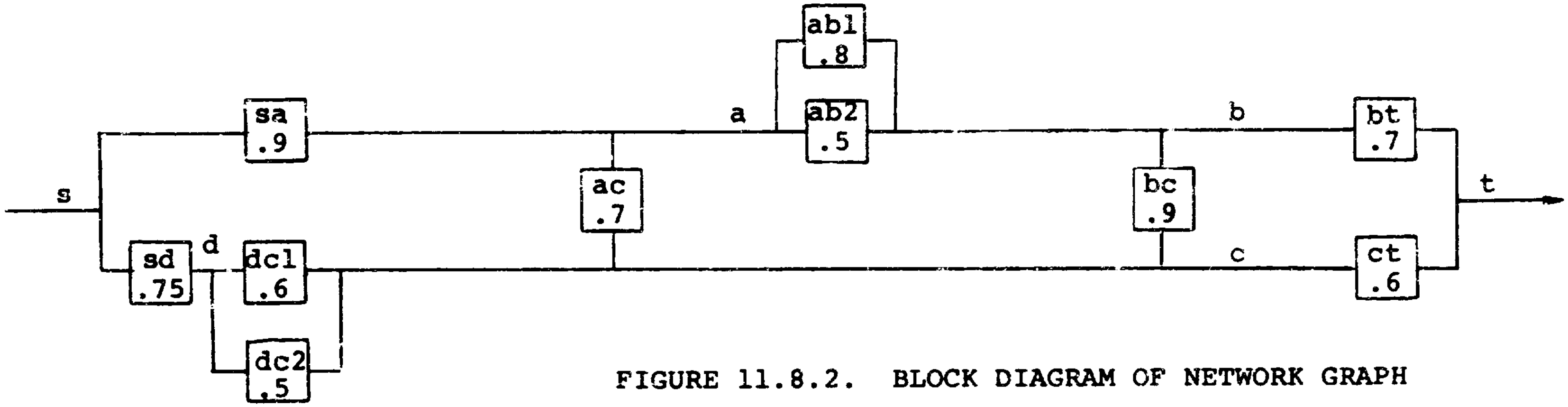


FIGURE 11.8.2. BLOCK DIAGRAM OF NETWORK GRAPH

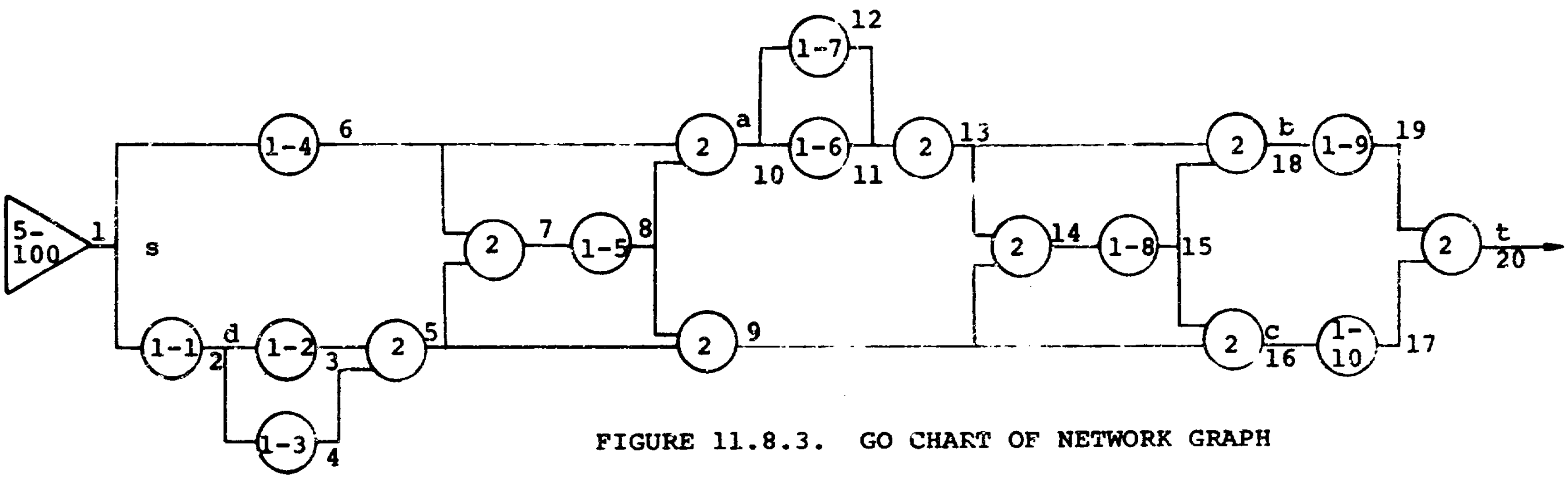


FIGURE 11.8.3. GO CHART OF NETWORK GRAPH

Table 11.8.1 contains the input data deck for this example.

TABLE 11.8.1. GO OPERATOR, KIND & PARAMETER DATA FOR EXAMPLE 8.

```
GO APPLICATIONS EXAMPLE 8 - NETWORK ANALYSIS
 $PARAM VALUES=2$
5 100 1$
1 1 1 2$
1 2 2 3$
1 3 2 4$
2 0 2 3 4 5$
1 4 1 6$
2 0 2 5 6 7$
1 5 7 8$
2 0 2 5 8 9$
2 0 2 6 8 10$
1 7 10 12$
1 6 10 11$
2 0 2 11 12 13$
2 0 2 9 13 14$
1 8 14 15$
2 0 2 9 15 16$
2 0 2 13 15 18$
1 10 16 17$
1 9 18 19$
2 0 2 17 19 20$
0 20$
EOR
GO APPLICATIONS EXAMPLE 8 - NETWORK ANALYSIS
1 1 .75 .25$
2 1 .6 .4$
3 1 .5 .5$
4 1 .9 .1$
5 1 .7 .3$
6 1 .5 .5$
7 1 .8 .2$
8 1 .9 .1$
9 1 .7 .3$
10 1 .6 .4$
100 5 1 0 1.$
EOR
GO APPLICATIONS EXAMPLE 8 - NETWORK ANALYSIS
 $PARAM PMIN=1.E-10$
EOR
```

The results for this example are contained in Table 11.8.2.

TABLE 11.8.2. RESULTS FOR EXAMPLE 8.

```
FINAL EVENT TABLE (INFINITY = 1)

                         SIGNALS AND THEIR VALUES

PROBABILITY       20
.1692928000        1
.8307072000        0
----------------------------------------------

TOTAL PROBABILITY = 1.0000000000
TOTAL ERROR =        .0000000000

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.    20
 0     .8307072000
 1     .1692928000
```

The probability that information can be transferred successfully from node s to node t is 0.8307072. The probability that it will not be under the conditions specified is 0.1692928. This is the exact result obtained in Reference 17.

11.9 Example 9 - Servo Power Supply [18]

The purpose of this example is to demonstrate how to conduct sensitivity studies using the GO methodology. Two meanings of the word sensitivity are employed here. The first meaning indicates the rate of change of system reliability with respect to change in component reliability, i.e., the partial derivative. The second meaning is the potential enhancement that could be achieved in system reliability by making a component perfect.

Consider the schematic diagram of the Servo Power Supply for the NASA Standard Tape Recorder as shown in Figure 11.9.1. The purpose of the servo power supply is to convert 16.5V dc power to five other levels of dc power - 24V, 14V, 7V, -14V, and -18V.

The operation of standard dc-to-dc converters was described in the April 1st, 1976 issue of "Electronics" magazine as follows.

> "The push-pull dc-dc converter is actually a free-running oscillator that produces an unregulated square-wave output. The dc input is chopped into complementary square waves, passed through a transformer, then rectified and filtered.
>
> During each cycle, the transistors are driven between cutoff and saturation through the positive feedback windings of the transformer primary. For half a cycle, the voltage across the transformer primary stays constant, as the magnetizing flux steadily increases. When the transformer core saturates, the magnetizing current rises suddenly to its peak value, the rate of rise of the magnetizing flux decays to zero, and the voltage across the primary windings collapses. This removes the base drive to the saturated transistor, turning it off."

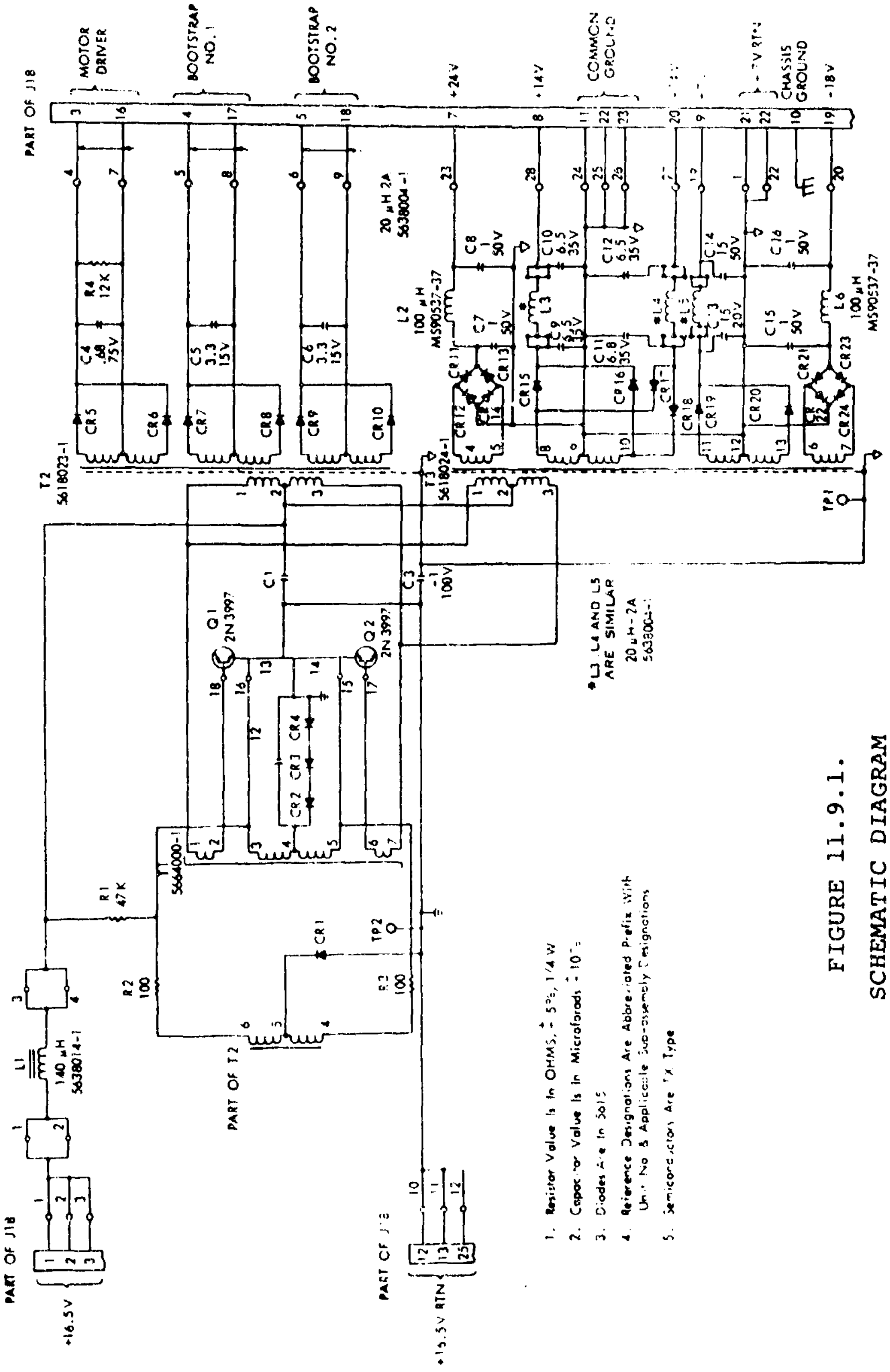


FIGURE 11.9.1.
SCHEMATIC DIAGRAM
SERVO POWER SUPPLY.

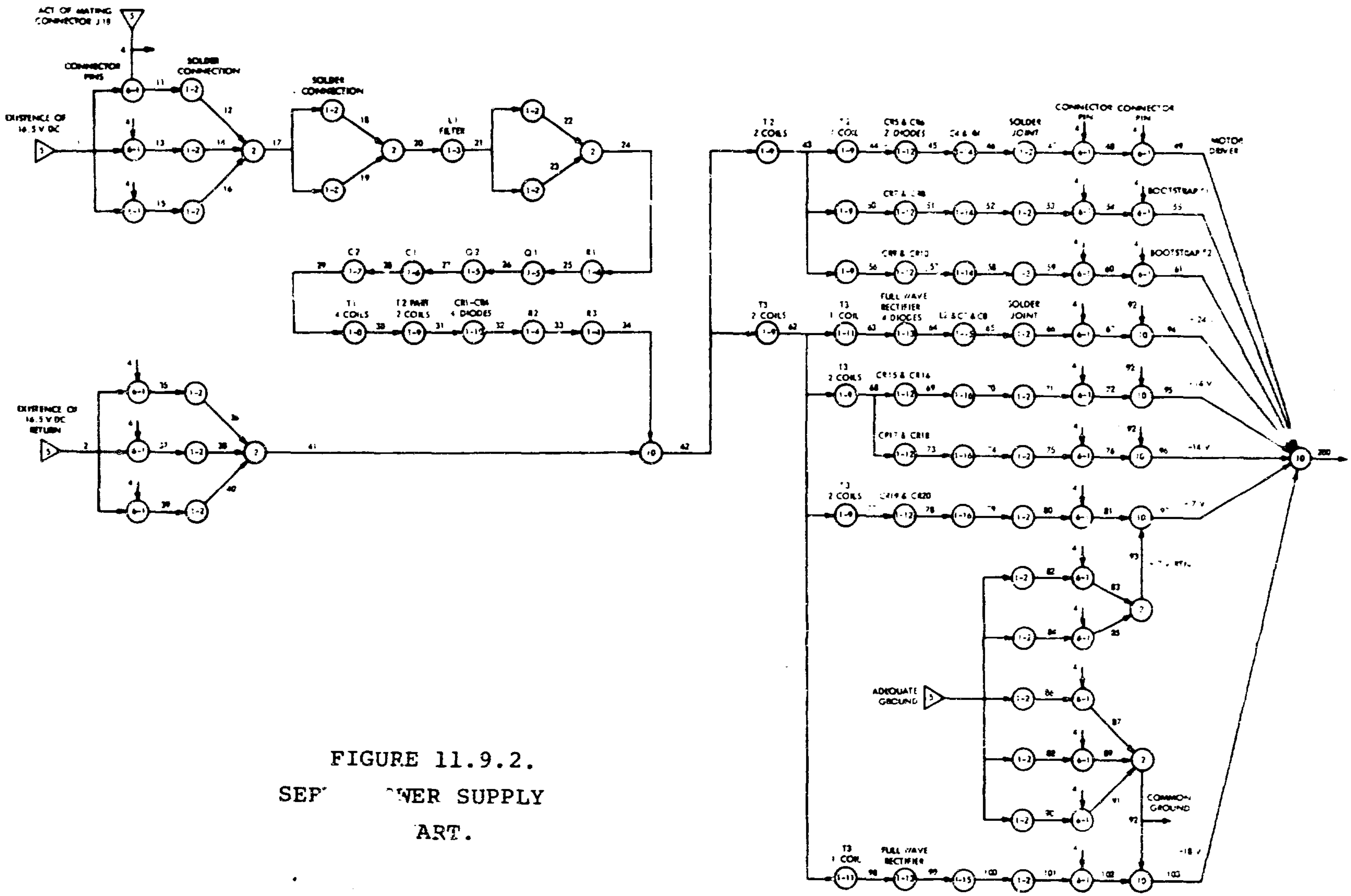


FIGURE 11.9.2.
SEP' 'WER SUPPLY
'ART.

With the electrical schematic and an understanding of the constituent components - connectors, filters, diodes, resistors, transistors, capacitors, transformers and solder joints - a GO model of the system was prepared as shown in Figure 11.9.2. The logic flow parallels that of the electrical schematic commencing with the triangular type 5 component depicted as signal 1 representing the existence of 16.5 V dc input power in the upper left and terminating with the type 10 "and" operator creating signal 200 on the far right.

Table 11.9.1 records the input data used for this example. The kind probabilities listed were arbitrarily chosen reflecting Safeguard missile data for first operation of systems built and tested to rigorous specifications. The initial results are documented in Table 11.9.2 indicating a system reliability of 0.9954.

If more visibility concerning particular outputs were desired, their signal numbers could have been retained and been explicit in the output array. For example, if the motor driver output probabilities of operation were desired explicitly, signal 49 could be retained and the output array would appear as shown in Table 11.9.3.

In this case the probability of proper motor driver power is shown to be 0.9985. The likelihood that signal 49 is adequate at the same time that one or more of the other power outputs is not is 3.1069×10^{-3}. From these joint distributions desired reliability information about various combinations of events can be readily obtained.

In the prior calculations the object was to determine the probability that the Servo Power Supply would provide proper output power when actuated. To discuss its

TABLE 11.9.1. GO OPERATOR, KIND & PARAMETER DATA FOR EXAMPLE 9.

```
GO APPLICATION EXAMPLE 9 - SERVO POWER SUPPLY
 $PARAM INFIN=3$
100 -1 101 201 202$
5 21 201$
6 1 201 101 11$
6 1 201 101 13$
6 1 201 101 15$
1 2 11 12$
1 2 13 14$
1 2 15 16$
2 0 3 12 14 16 202$
EOR
101 -1 102 103 203$
1 9 102 44$
1 12 44 45$
1 14 45 46$
1 2 46 47$
6 1 47 103 48$
6 1 48 103 203$
EOR
102 -1 104 105 106 204$
1 11 104 63$
1 13 63 64$
1 15 64 65$
1 2 65 66$
6 1 66 105 67$
10 0 2 67 106 204$
EOR
103 -1 107 108 109 205$
1 12 107 69$
1 16 69 70$
1 2 70 71$
6 1 71 108 72$
10 0 2 72 109 205$
EOR
5 20 4$
100 0 4 1 17$
1 2 17 18$
1 2 17 19$
2 0 2 18 19 20$
1 3 20 21$
1 2 21 22$
1 2 21 23$
2 0 2 22 23 24$
1 4 24 25$
```

TABLE 11.9.1. (Continued)

```
1 5 25 26$
1 5 26 27$
1 6 27 28$
1 7 28 29$
1 8 29 30$
1 9 30 31$
1 10 31 32$
1 4 32 33$
1 4 33 34$
100 0 4 2 41$
10 0 2 34 41 42$
1 9 42 43$
101 0 43 4 49$
101 0 43 4 55$
101 0 43 4 61$
1 9 42 62$
100 0 4 3 92$
102 0 62 4 92 94$
102 0 62 4 92 103$
1 9 62 68$
103 0 68 4 92 95$
103 0 68 4 92 96$
1 9 62 77$
1 2 3 82$
1 2 3 84$
6 1 82 4 83$
6 1 84 4 85$
2 0 2 83 85 93$
103 0 77 4 93 97$
10 0 8 49 55 61 94 95 96 97 103 200$
0 200$
EOR
GO APPLICATION EXAMPLE 9 - SERVO POWER SUPPLY
1 6 0.999999 0.000001 0.0 $CONNECTOR PINS
2 1 0.999999 0.000001 $SOLDER JOINTS
3 1 0.99995 0.00005 $L1 FILTER
4 1 0.999999 0.000001 $RESISTORS
5 1 0.9999 0.0001 $TRANSISTORS Q1,Q2
6 1 0.9999 0.0001 $CAPACITOR C1
7 1 0.9999 0.0001 $CAPACITOR C2
8 1 0.9998 0.0002 $T1 - 4 COILS
9 1 0.9999 0.0001 $T2 - 2 COILS
10 1 0.9998 0.0002 $4 DIODES
11 1 0.99995 0.00005 $T3 - 1 COIL
12 1 0.9999 0.0001 $2 DIODES
13 1 0.9998 0.0002 $4 DIODES - FULL WAVE RECTIFIER
14 1 0.9999 0.0001 $CAPACITOR - RESISTOR FILTER
```

TABLE 11.9.1. (Co[illegible]

```
15 1 0.99975 0.00025 $2 CAPACITORS AND COIL FILTER L2, L6
16 1 0.9997 0.0003 $2 CAPACITORS, COIL, 2 LOGIC CHIPS, L3,L4,L5
20 5 2 2 0.9999 3 0.0001$
21 5 1 1 1.$
EOR
GO APPLICATION EXAMPLE 9 - SERVO POWER SUPPLY
 $PARAM PMIN=1.E-10$
EOR
```

TABLE 11.9.2. GO RESULTS FOR EXAMPLE 9.

FINAL EVENT TABLE (INFINITY = 3)

PROBABILITY	SIGNALS AND THEIR VALUES 200
.0045619377	3
.9954380461	2

TOTAL PROBABILITY = .9999999838
TOTAL ERROR = .0000000162

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	200
2	.9954380461
3	.0045619377

TABLE 11.9.3. MOTOR DRIVER RELIABILITY.

PROBABILITY	SIGNAL NUMBERS AND TIMES 49	200
.0014550222	3	3
.0031069155	2	3
.9954380461	2	2

reliability as a function of time, given that it was initially operational, the degradations of the various components with time are needed. This relationship is typically a function of the stress loadings and environmental use conditions.

The failure rate data utilized in assessing the characteristics of the continuously operating servo power supply was extracted from Mil-Handbook 217B using the lowest available temperature stress levels. This was an arbitrary decision and refined analyses would define the use environment as precisely as possible.

The data extracted from Mil-Handbook 271B is recorded in Table 11.9.4 along with component reliability estimates for 10^5, 5×10^5 and 10^6 hours based on the failure rates noted. Using this and similar information for other points in time the Servo Power Reliability curve as a function of time was drawn from the results of repeated GO runs, as shown in Figure 11.9.3.

The curve represents the probability that all power outputs remain within prescribed bounds as a function of time, given that it performed properly at initialization. Using the method defined in the document, A Procedure to Derive MTBF for Complex Systems Using GO, K-76-42U(R), the MTBF of this system was determined to be 4.05×10^5 hours using the data noted.

This is a relatively long time, 46.2 years, but if the environment is benign and the component failure rates accurate this would be a valid estimate for the system MTBF. In practice, the effects of aging, random shocks, radiation, vibration, temperature cycling and extremes are often not properly incorporated into the failure rate estimates.

TABLE 11.9.4. SERVO COMPONENT RELIABILITY DATA.

KIND NO.	FAILURE RATE λ	RELIABILITY 10^5 HOUR	RELIABILITY 5×10^5 HOURS	RELIABILITY 1×10^6 HOURS	COMPONENT DESCRIPTION
6-1	-	0.999999	0.999999	0.999999	Connector pins
1-2	-	0.999999	0.999999	0.999999	Solder joints, board holes
1-3	2×10^{-9}	0.999980	0.99900	0.99800	L1 Filter
1-4	2×10^{-7}	0.98030	0.90484	0.81873	Resistors
1-5	1×10^{-8}	0.99900	0.99501	0.99005	Transistors Q1, Q2
1-6	6×10^{-8}	0.99402	0.97045	0.94176	Capacitor C1
1-7	6×10^{-8}	0.99402	0.97045	0.94176	Capacitor C2
1-8	2.8×10^{-9}	0.99972	0.99860	0.99720	T1 - 4 coils
1-9	1.4×10^{-9}	0.99986	0.99930	0.99860	T2 - 2 coils
1-10	2×10^{-8}	0.99800	0.99005	0.98020	4 Diodes
1-11	7×10^{-10}	0.99993	0.99965	0.99930	T3 - 1 coil
1-12	1×10^{-8}	0.99900	0.99501	0.99005	2 Diodes
1-13	2×10^{-8}	0.99800	0.99005	0.98020	4 Diodes - full wave rectifier
1-14	2.6×10^{-7}	0.98142	0.86502	0.74826	Capacitor - resistor filter
1-15	1.4×10^{-7}	0.98610	0.93239	0.86936	2 Capacitors & coil, filter L2, L6
1-16	2.0×10^{-7}	0.98030	0.90484	0.81873	2 Capacitors 2 logic chips - filters L3, L4, L5
System	2.46×10^{-6}	0.78596	0.30005	0.09004	

System MTBF $= 4.05\times10^5$ hours

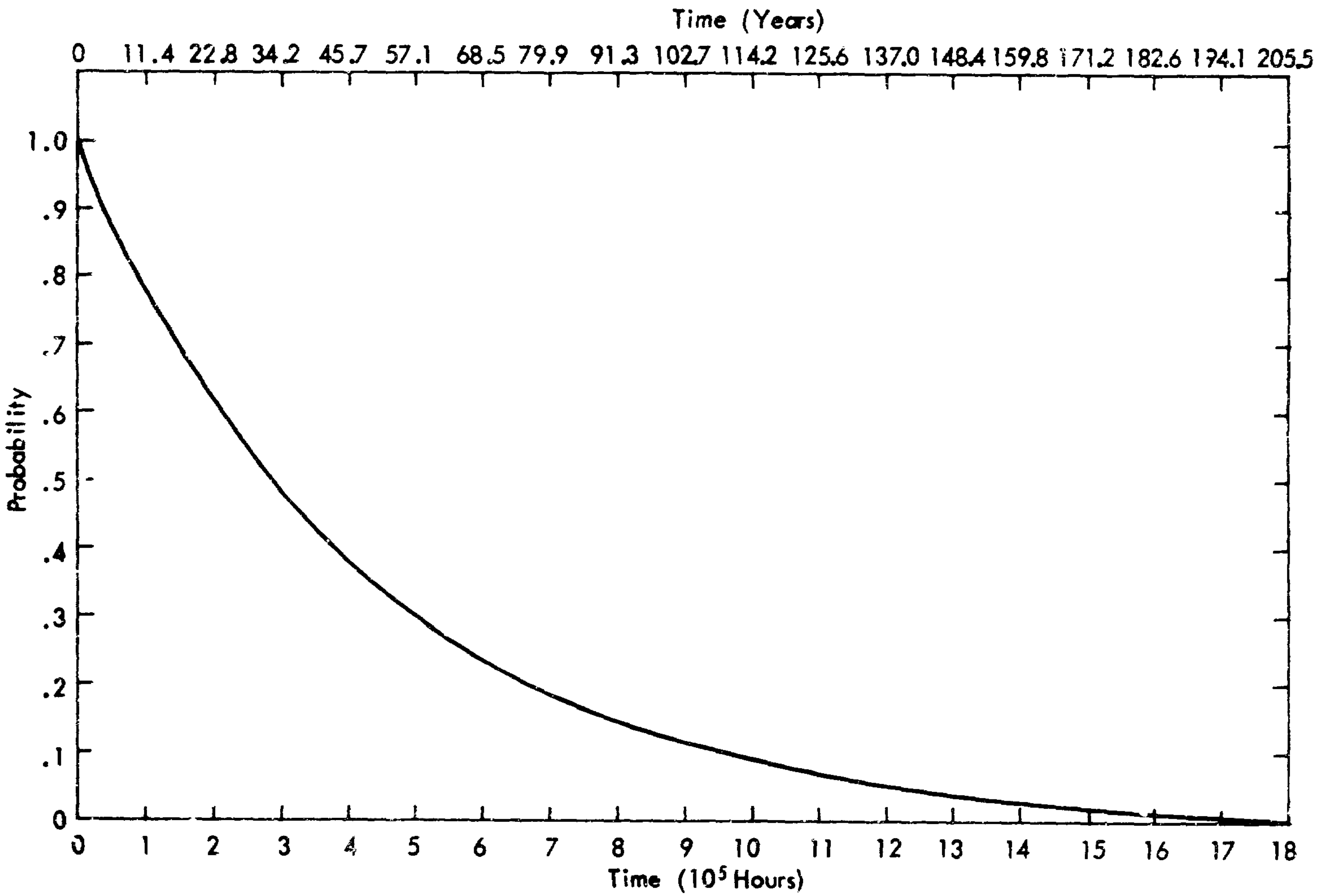


FIGURE 11.9.3. SERVO POWER SUPPLY RELIABILITY.

These results appear to be somewhat optimistic even though the Servo Power Supply is a relatively simple system using standard, well-known components.

Often the question is asked, if reliability enhancement were desired how could it be achieved. If redesign were an option it would be relatively easy to suggest alterations, modify the GO model appropriately and systematically compare the calculated results.

If no redesign were contemplated, a sensitivity study is typically used to determine the rates of change of system reliability to changes in component reliability and to deduce the maximum enhancement that could be achieved by component improvement. This is done for one specific point in time. The results of such an exercise often determine desirable design modifications.

A sensitivity study for this system was performed with the results shown in Table 11.9.5. The Δr column indicates the change in component reliability and the ΔR column the resultant effect on the system. The ratio, $\Delta R/\Delta r$, is essentially the partial derivative and indicates the rate of change of system reliability to component reliability.

Note that with the exception of kinds 1 and 2, the connector pins and solder joints, that the slopes correspond to the numbers of components. This documents the fact that each of these components must function properly for system success as it has been defined, i.e., requiring that output power on all lines must be within prescribed limits.

The slope of 11 for the connector pins indicates that because of redundancy 11 of the connector pins are not crucial to successful system operation. Similarly for the

solder joints, 15 are redundant. These results are precisely what a capable engineer would know. For large systems with multiple redundancies and cross connections such effects are not always obvious. The GO procedure, once a system is correctly modeled, is unerring.

The component reliability changes in Table 11.9.4 were selected to make the components perfect. Hence, the effect on the system, the R column entries, document the total system reliability enhancement that could be achieved by enhancing the components individually.

By examining the ΔR column the largest increment in system reliability could be achieved by enhancing the L3, L4, L5 filters (kind 16) to perfection . The next best enhancement would be the T2 coils, (kind 9) etc. Note that even though the rate of change of system to component reliability is greatest for connector pins and solder joints that their reliabilities are so high little enhancement can be achieved by concentration on their improvement.

TABLE 11.9.5. SENSITIVITY STUDY.

KIND NO.	Δr	ΔR	$\frac{\Delta R}{\Delta r}$	NO. OF COMPONENTS
1	1×10^{-6}	1.1×10^{-5}	11	22
2	1×10^{-6}	8×10^{-6}	8	23
3	5×10^{-5}	5×10^{-5}	1	1
4	1×10^{-6}	3×10^{-6}	3	3
5	1×10^{-4}	2×10^{-4}	2	2
6	1×10^{-4}	1×10^{-4}	1	1
7	1×10^{-4}	1×10^{-4}	1	1
8	2×10^{-4}	2×10^{-4}	1	1
9	1×10^{-4}	8×10^{-4}	8	8
10	2×10^{-4}	2×10^{-4}	1	1
11	5×10^{-4}	1×10^{-4}	2	2
12	1×10^{-4}	6×10^{-4}	6	6
13	2×10^{-4}	4×10^{-4}	2	2
14	1×10^{-4}	3×10^{-4}	3	3
15	2.5×10^{-4}	5×10^{-4}	2	2
16	3×10^{-4}	9×10^{-4}	3	3

11.10 Example 10 - Feedback Loop Analysis - Diesel Generator [19]

The purpose of this example is to demonstrate how to properly model feedback loops, i.e., where the output of an equipment influences its input. We will introduce the concept of replicated component supertypes to handle this modeling problem.

11.10.1 Diesel Generator System

Consider the abbreviated schematic diagram of Figure 11.10.1 showing a portion of the starting control system of an emergency diesel generator.

The actuating elements shown in Figure 11.10.1 are:

a. TRIP: The engine starting signal; this could be either an automatic or a manual action.

b. ASR: Auto-Start Relay.

c. STR: Starting Relay.

d. TD2: Time Delay Relay; the function of this relay is to terminate the attempted start if the engine does not reach a certain critical speed within a specified time.

e. AST: Air Start Solenoid; this device causes compressed air to be sent to the air-powered starter which in turn starts the diesel itself; the air starter is not explicitly shown on the diagram but may be considered as being included in the ENGINE.

f. ES: Engine Start; this device is actuated when the engine speed reaches the critical value mentioned above.

g. ESR: Engine Start Relay; this relay deactuates the air starter and (not shown) is used to apply the electric field to the generator.

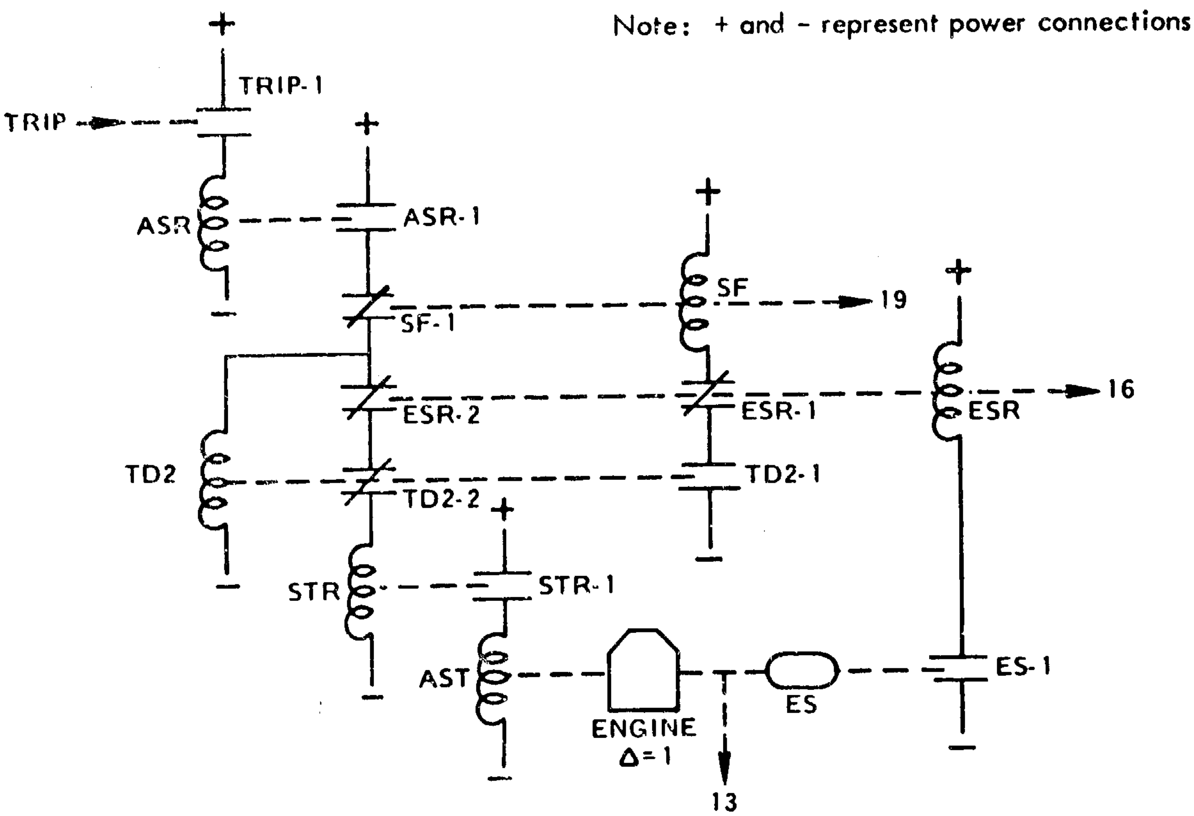


FIGURE 11.10.1. CONTROL SYSTEM SCHEMATIC.

h. SF: Start Failure Relay; in the event of a failure to start within the prescribed time, this relay will deactuate the air starter and (not shown) actuate an alarm system.

The symbols + and - indicate the positive and negative power connections to the various actuators. The notation ESR-1 names contact #1 of ESR, etc.

Using GO time values, the normal operation of the system is as follows:

t = 1: Power is applied.

t = 2: The TRIP signal arrives; this will actuate the air starter and also TD2; TD2 will function at t=4 after a delay of $\Delta = 2$.

t = 3: The ENGINE will reach its certain speed, and ES will be actuated which in turn will actuate ESR.

Note that in normal operation TD2 accomplishes nothing and SF will never be actuated.

For the purposes of our analysis, the events of interest are the occurrence times of the actuation of the ENGINE, ESR, and SF; these events are represented by the GO signals 13, 16 and 19 respectively and are indicated on Figure 11.10.1.

11.10.2 The Modeling Problem

Inspection of Figure 11.10.1 reveals a feedback loop starting at ES and going back to the contacts SF-1 and ESR-2. In addition TD2 enters this loop via TD2-1. The five components ES, ES-1, ESR, ESR-1, and SF are normally controlled by signal 13, the ENGINE output; but an actuation of SF-1 by a premature signal from SF will affect signal 13. Similarly the input to TD2 can be affected by a premature actuation of TD2 itself.

The key to a GO modeling solution to the feedback problem is the recognition that in essence each component which lies within a feedback loop must be "looked at" twice. The first look must occur before the termination point of the loop (SF-1 in this case) is analyzed and the second look must occur after the beginning point(s) of the loop (TD2 and ES) are reached. Thus, in our example, the GO analysis must proceed in approximately the following sequence (after ASR-1 is finished): ES, ES-1, ESR, TD2, TD2-1, ESR-1, SF (this gets us through the loop initially), SF-1, ESR-2, TD2, TD2-2, STR, STR-1, AST, ENGINE, ES, ES-1, ESR, TD2-1, ESR-1, and finally SF.

A first (and incorrect) solution to this "double-look" problem is simply to replicate the loop operators. Thus, for example, we would represent ESR in its two occurrences by two type 3 operators, each having the same operational mode distribution - that is, the same good, bad, and premature probabilities. The error in this method stems from the fact that in GO distinct operators always have statistically independent mode distributions, and consequently we would, for example, be allowing ESR to premature and be normal simultaneously - a situation which is clearly erroneous. Nevertheless, this idea contains the germ of a correct method.

The solution which we propose (there may of course be others) is to break an operator into two parts. The first part will be a type 13 operator (the driver) with one or two outputs and no inputs. The operational mode distribution of the component will be reflected in the values of the type 13 outputs (the driving signals). The second part will be an appropriate supertype which contains only perfect operators

and which when controlled by the driving signals will produce the same operational results as the original unmodified operator. This second part, which we will call a type replica, can then be introduced at several points in a GO analysis sequence, and correct results will be obtained because of the perfect (non-stochastic) nature of the replica. The driving signals must, of course, be introduced before the first replica and retained until the last one (this retention can introduce some problems - see Section 11.10.4). Using this scheme, downstream operator responses are properly conditioned by prior logical usages of given components.

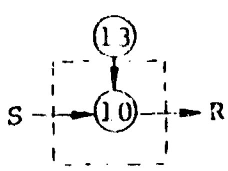
11.10.3 Type Drivers and Replicas

The drivers and replicas for type 1, 3, 6, 7, and 8 operators are defined in this section. Other modelings are undoubtedly possible, but the ones given here function correctly and are basically quite simple (we are not well satisfied with the type 7 however).

11.10.3.1. Type 1 Driver and Replica

Replace $S \longrightarrow (1) \longrightarrow R$

by



where

Component Mode	Probability	Value of Signal 1
good	g	0
bad	b	∞

The type 13 operator is the driver, and its kind data is:

K 13 0 1 1 2 0 g ∞ b $

Alternatively the driver can be a type 5 with kind data

K 5 2 0 g ∞ b $

For other types (see below) type 13 drivers are required, and so for the sake of consistency a type 13 may be preferred here instead of the slightly simpler type 5.

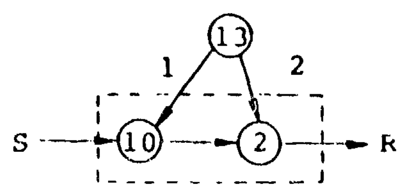
The type 10 operator enclosed in dashed line constitutes the replica. This might be modeled as a supertype for the sake of consistency with other type replicas.

The action of the driver-replica combination is quite straightforward. If the component is good (with probability g), the value of signal 1 is 0, and consequently the value of the response (R) will equal that of the stimulus (S). If the component is bad (with probability b), the value of signal 1 is ∞ regardless of the value of S.

11.10.3.2 Type 3 Driver and Replica

Replace $S \longrightarrow (3) \longrightarrow R$

by



where

Comp. Mode	Probability	Values of Signals 1	2
good	g	0	∞
bad	b	∞	∞
prem.	p	∞	0

The kind data for the type 13 driver is

K 13 0 2 1 3 0 ∞ q ∞ ∞ b ∞ 0 p $

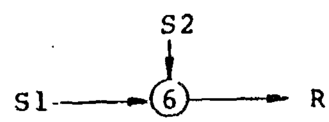
where K will of course be replaced by the appropriate kind number.

The replica within the dashed line will normally be modeled as a supertype. Note that this same replica supertype may be used for several different type 3's within a problem as well as for several replications of a particular type 3 because the replica is perfect; this comment applies to replicas of all types.

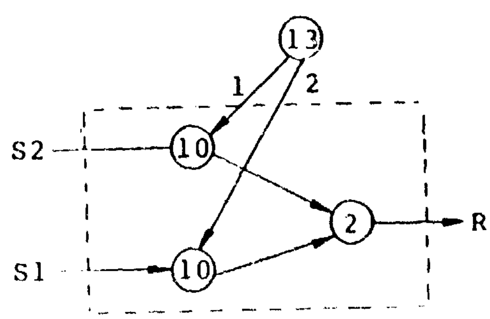
We will leave it to the reader to verify the correctness of the driver-replica model.

11.10.3.3 Type 6 Driver and Replica

Replace



by



where

Comp. Mode	Probability	Value of Signals 1	2
good	g	0	∞
bad	b	∞	∞
prem.	p	∞	0

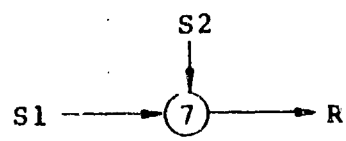
The kind data for the type 13 driver is

K 13 0 2 1 3 0 ∞ g ∞ ∞ b ∞ 0 p $

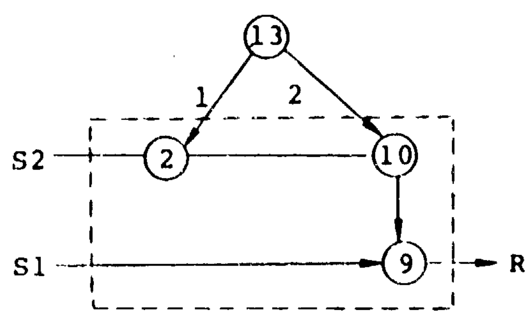
Note that the kind data is the same as for the type 3 driver and consequently may be used for both types in the same problem if the mode probabilities are the same.

11.10.3.4 Type 7 Driver and Replica

Replace



by



where

Comp. Mode	Probability	Value of Signals 1	2
good	g	∞	0
bad	b	0	∞
prem.	p	0	0

The kind data for the type 13 driver is

K 13 0 2 1 3 ∞0 g 0 ∞ b 0 0 p $

and for the type 9 is

K 9 ∞+1 0 0 1 0 ... ∞0 $

The type 9 may be replaced by a perfect type 7. The type 9 is a bit more efficient for GO3, but the type 7 would be simpler for the analyst.

Note that the driver data differs from that of the type 3 and type 6. It would be desirable to have a replica which used the latter form, but we have been unsuccessful in creating a model which works correctly with that form.

3.2.5 <u>Type 8</u>

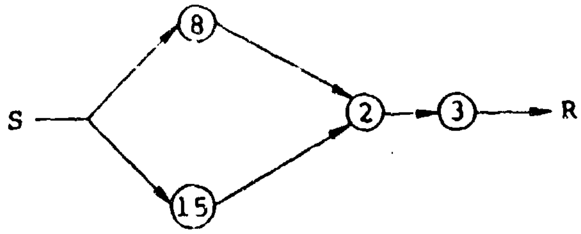
11.10.3.5 Type 8 Driver and Replica

A nonreplicated imperfect type 8 can be modeled by

$$S \longrightarrow 8 \longrightarrow 3 \longrightarrow R$$

where the premature and bad modes are incorporated into the type 3, and type 8 simply provides the increment.

Caution: the above representation is not correct if an input value of 0 is possible and it is desired to have such an input produce an output of 0 regardless of the increment - for example, if one wishes to interpret time C as an indefinite period preceding time 1. This situation can be properly modeled by:



where the type 15 kind data is

K 15 0 ∞ 0 0 0. 1. $

and provides an output from the type 15 of 0 if the input is 0 and ∞ otherwise. The output of the type 2 will then be 0 if the input S has a value of 0, and the value of S increased by the type 8 increment otherwise.

For a type 8 with a single increment a replica can be modeled by using a standard type 3 driver and a replica as shown by one of the above where the type 3 is replaced by a type 3 replica.

If the type 8 has more than one increment, a different model will be necessary, but we have not investigated this as yet.

11.10.4 Signal Retention and Feedback Loop Identification

The necessity of retaining the driving signals for application to two (or more) operator replicas is a potentially limiting feature. If a large number of these signals must be simultaneously retained, the number of terms in the intermediate probability distributions may become extremely large, and this can have a severe impact on both program execution time and accuracy because of the necessity of extensive pruning in order to keep the distribution size

down to manageable values. Beyond exercising some care in the modeling process, there is little that can be done to ease this problem, but the analyst should be aware of the limitation. In several applications to date the introduction of 15-20 such signals has not been restrictive.

The identification of the boundaries of the feedback loop(s) in a system may present some problems. Our experience at this point is quite limited, and we have not generated definitive rules. Caution is obviously required.

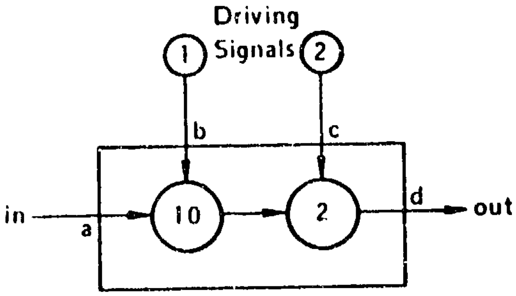
11.10.5 Example Diesel Generator Control System Analysis

In this section we show a GO model for the simplified control system described earlier (see Figure 11.10.1 for the schematic diagram) and give the results of the GO analysis.

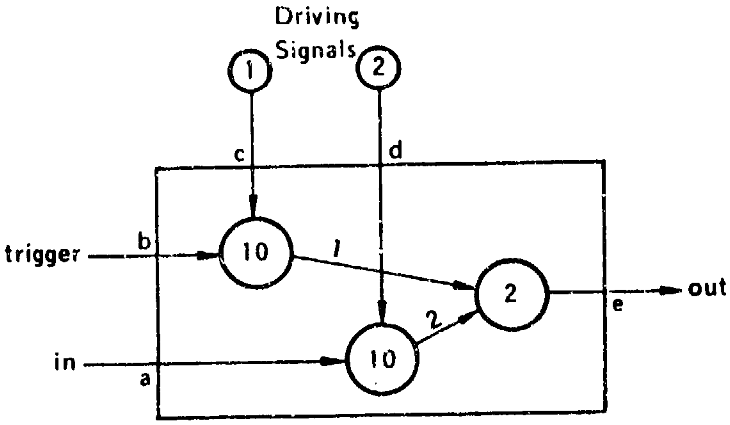
In the model power is applied (signal 1) at time t=1 with probability 1 and the TRIP signal (signal 2) at t=2 with probability 1. Infinity (never) is at t=7.

The final signals are 13 (engine up to speed), 16 (ESR actuation), and 19 (SF actuation). With normal operation these signals should occur at times 3, 3, and 7 ($=\infty$) respectively.

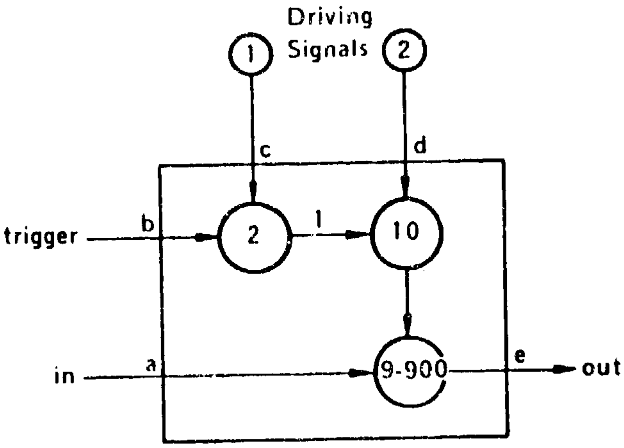
Figure 11.10.2 shows the replica supertypes. Note the use of lower case letters to indicate the various inputs and outputs of each supertype. By this means the use of the signals in the main GO chart can readily be identified. Also note that the simplified version (without the type 15) of the type 8 replica is used; this is possible because the TD2 input comes from the power signal and cannot occur before time 1.



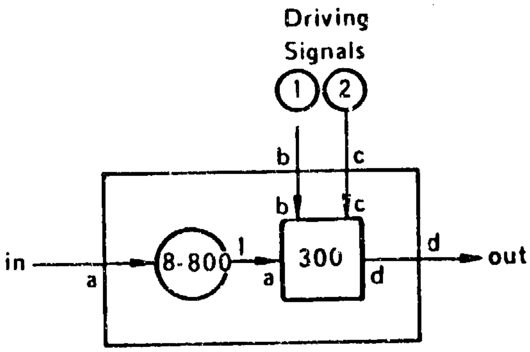
ST 300
(Type 3 replica)



ST 600
(Type 6 replica)



ST 700
(Type 7 replica)



ST 800
(Type 8 replicas; $\Delta = 2$)

FIGURE 11.10.2. SUPERTYPE DEFINITIONS.

The main GO chart is shown in Figure 11.10.3. On this chart dashed signal lines indicate that the signal is used at some other point(s); this convention is slightly inconvenient but avoids cluttering up the drawing with connecting lines.

Each operator symbol includes the type, kind, name (for correlation with the schematic), and operator number(s) (as assigned by GO1 and for use both in identifying the analysis sequence and in identifying the operators involved in the fault sets produced by the Fault Finder). The operator names are coded in the following manner (ESR is used as an example):

Nonreplicated operator:

ESR	:	ESR actuator.
ESR-1	:	contact #1 of ESR.

Replicated operators:

ESR/	:	driver (type 13) for ESR actuator.
ESR/1	:	1st replica of ESR actuator.
ESR-1/	:	driver for contact #1 of ESR.
ESR-1/2	:	2nd replica of contact #1 of ESR.

The GO chart of Figure 11.10.3 can be almost directly overlaid on the schematic of Figure 11.10.1. The six supertype replicas at the right side of the chart can be mentally shifted to be left and overlaid on their counterparts which are analyzed first.

The GO1 output (reflecting the input operator data) is shown in Table 11.3. Note the maximum number (18) of simultaneously active signals. This suggests a relatively large error may occur in GO3.

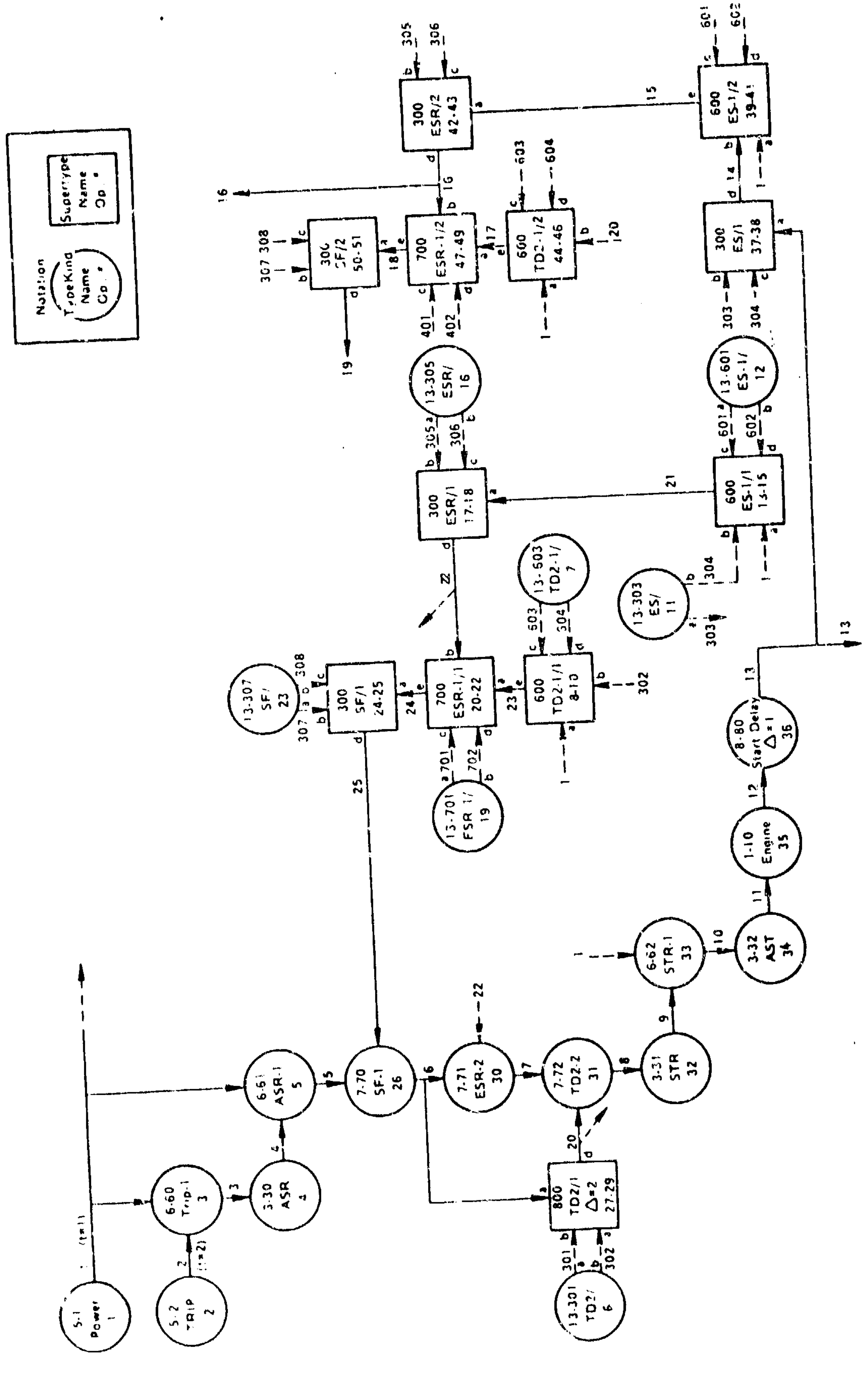


FIGURE 11.10.3. CONTROL SYSTEM GO CHART.

The GO2 output is shown in Table 11.10.2 reflecting the probabilistic data specified. In particular note the probabilities associated with the type 13 operators which document the premature, success and dud probabilities of the components being replicated.

Table 11.10.3 shows the GO3 output. Note the total error of 4.17×10^{-6} which has been influenced by the required signal retention. This effect is also revealed in the large intermediate distributions. The error could, of course, be reduced by choosing a smaller PMIN, but this would have resulted in appreciably larger distributions and longer execution times (the GO3 execution time here was 3.7 sec).

Recall that signals 13, 16, and 19 represent the outputs from the ENGINE, ESR and SF equipments respectively. The most likely event is therefore the success event with the ENGINE being started successfully at time point 3 and the SF relay failing to start as it should if the ENGINE is up to speed.

The second most likely event is that both the ENGINE and the ESR relay fail and the SF relay is energized at time point 4, etc. All possible system states are identified and this system involving the two specified feedback loops has thus been correctly modeled and its probabilistic responses characterized.

The capability to properly model and analyze systems with feedback loops has thus been demonstrated. This novel application of the GO procedure is another powerful tool enhancing its generality and versatility.

```
   RUN ON 07/22/77 AT 14.00.13.
VALUES = 8, BIAS = 5000, OPS = 1, SIGNALS = 0, ERRORS = 25
OPERATOR DATA
OP    DATA
----  ----------------------------------------------------------

XXXX  300 -1 101 102 103 201 $ TYPE 3 REPLICATION
XXXX  10 0 2 101 102 1 $
XXXX  2 0 2 1 103 201 $
----  END OF SUPER TYPE 300
XXXX  600 -1 101 102 103 104 201 $  TYPE 6 REPLICATION
XXXX  10 0 2 102 103 1 $
XXXX  10 0 2 101 104 2 $
XXXX  2 0 2 1 2 201 $
----  END OF SUPER TYPE 600
XXXX  700 -1 101 102 103 104 201 $  TYPE 7 REPLICATION
XXXX  2 0 2 102 103 1 $
XXXX  10 0 2 1 104 2 $
XXXX  9 900 101 2 201 $
----  END OF SUPER TYPE 700
XXXX  800 -1 101 102 103 201 $  TYPE 8 REPLICATION
XXXX  8 800 101 1 $
XXXX  300 0 1 102 103 201 $
----  END OF SUPER TYPE 800
   1  5 1 1 $ POWER
   2  5 2 2 $ TRIP
   3  6 60 1 2 3 $ TRIP-1
   4  3 30 3 4 $  ASR
   5  6 61 1 4 5 $  ASF-1
   6  13 301 0 2 301 302 $  TD2/
   7  13 603 0 2 603 604 $ TR2-1/
$$$$  600 0 1 302 603 604 23 $ TD2-1/1
   8  (L=1)      10     0     2   302  603 5001
   9  (L=1)      10     0     2     1  604 5002
  10  (L=1)       2     0     2  5001 5002   23
  11  13 303 0 2 303 304 $ ES/
  12  13 601 0 2 601 602 $  ES-1/

$$$$  600 0 1 304 601 602 21 $  ES-1/1
  13  (L=1)     10     0     2   304   601 5003
  14  (L=1)     10     0     2     1   604 5004
  15  (L=1)      2     0     2 5003 5004    21
  16  13 305 0 2 305 306 $  ESR/
XXXX  300 0 21 305 306 22 $  ESR/1
  17  (L=1)      10    0    2    21   305   5005
  18  (L=1)       2    0    2  5005   305     22
  19  13 701 0 2 701 702 $  ESR-1/1
XXXX  700 0 23 22 701 702 24 $  ESR-1/1
  20  (L=1)       2     0     2    22   701   5006
  21  (L=1)      10     0     2  5006   702   5007
```

TABLE 11.10.1.   G01 OUTPUT DATA.

TABLE 11.10.1 (Continued)

```
  22  (L=1)        9   900     23  5007      24
  23  13 307 0 2 307 308 $  SF/
XXXX  300 0 24 307 308 25 $  SF/1
  24  (L=1)      10     0     2     24    307 5008
  25  (L=1)       2     0     2   5008    308   25
  26  7 70 5 25 6 $  SF-1
XXXX  800 0 6 301 302 20 $  TD2/1
  27  (L=1)       8   800     6 5009
XXXX  (L=1)     300     0 5009  301   302    20
  28  (L=2)      10     0     2 5009   301 5010
  29  (L=2)       0     0     2 5010   302   20
  30  7 71 6 22 7 $ ESR-2
  31  7 72 7 20 8 $ TD2-2
  32  3 31 8 9 $  STR
  33  6 62 1 9 10 $  STR-1
  34  3 32 10 11 $  AST
  35  1  10  11  12  $  ENGINE
  36  8  80  12  13  $  ENGINE STARTING TIME
XXXX  300 0 13 303 304 14 $  ES/1
  37  (L=1)      10     0     2     13    303 5011
  38  (L=1)       2     0     2   5011    304   14
XXXX  600 0 1 14 601 602 15 $  ES-1/2
  39  (L=1)     10      0     2    14   601 5012
  40  (L=1)     10      0     2     1   602 5013
  41  (L=1)      2      0     2 5012 5013   15
XXXX  300 0 15 305 306 16 $ $  ESR/2
  42  (L=1)      10     0     2     15   305 5014
  43  (L=1)       2     0     2   5014   306   16
XXXX  600 0 1 20 603 604 17 $  TD2-1/2
  44  (L=1)      10     0     2    20   603   5015
  45  (L=1)      10     0     2     1   604   5016
  46  (L=1)       2     0     2 5015 5016     17
XXXX  700 0 17 16 701 702 18 $  ESR-1/2
  47  (L=1)       2     0     2    16   701   5017
  48  (L=1)      10     0     2 5017   702   5018
  49  (L=1)       9   900    17 5018    18
XXXX  300 0 18 307 308 19 $  SF/2
  50  (L=1)      10     0     2    18   307   5019
  51  (L=1)       2     0     2 5019   308     19
////  0 13 16 19 $

NUMBER OF OPERATORS     =   51
NUMBER OF SIGNALS       =   58
MAX NUMBER ACTIVE       =   18
MAX SIGNAL LIST SIZE    =   18
NUMBER OF SIGNAL VALUES=     8
NUMBER OF SIGNALS/WORD  =   28
NUMBER OF WORDS/TERM    =    1

FINAL SIGNALS =  13  16  19
```

```
FEEDBACK TEST
  RUN ON 07/22/77 AT 14.38.16.
OPERATOR FILE --- ( 07/22/77  14.06.13.)
RECORD   KIND DATA
------  --------------------------------------------------
     1   900 9 8 0 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 $  ST 700
     2   800 8 1 2 1 $  ST 800
     3   1 5 1 1 1 $  POWER
     4   2 5 1 2 1 $  TRIP
     5   80 8 1 1 1 $  ENGINE STARTING
     6   301 13 0 2 1 3 0 7 .997 7 7 .002 7 0 .001 $
     7   303 13 0 2 1 3 0 7 .997 7 7 .002 7 0 .001 $
     8   305 13 0 2 1 3 0 7 .997 7 7 .002 7 0 .001 $
     9   307 13 0 2 1 3 0 7 .997 7 7 .002 7 0 .001 $
    10   601 13 0 2 1 3 0 7 .997 7 7 .002 7 0 .001 $
    11   603 13 0 2 1 3 0 7 .997 7 7 .002 7 0 .001 $
    12   701 13 0 2 1 3 7 0 .997 0 7 .002 0 0 .001 $
    13   10 1 .998 .002 $
    14   30 3 .997 .002 .001 $
    15   31 3 .997 .002 .001 $
    16   32 3 .997 .002 .001 $
    17   60 6 .997 .002 .001 $
    18   61 6 .997 .002 .001 $
    19   62 6 .997 .002 .001 $
    20   70 7 .997 .002 .001 $
    21   71 7 .997 .002 .001 $
    22   72 7 .997 .002 .001 $
-----------------------------------------------------------

USE SUMMARY TABLE.  ENTRY = KIND/TYPE(FREQUENCY)
     (FREQUENCY IS NEGATIVE FOR PERFECT KINDS.)

   1/ 5(    1)   2/ 5(    1)  10/ 1(    1)  30/ 3(    1)  31/ 3(    1)
  70/ 7(    1)  71/ 7(    1)  72/ 7(    1)  80/ 8(    1) 301/13(    1)
 603/13(    1) 701/13(    1) 800/ 8(    1) 900/ 9(    2)

NUMBER OF KINDS INPUT ----   22
NUMBER USED - NONPERFECT -   22
NUMBER USED - PERFECT ----    0
```

TABLE 11.10.2.   GO2 OUTPUT DATA.

RUN ON 07/22/77 AT 14.11.07.

OPERATOR FILE --- (07/22/77 14.06.13.) FEEDBACK TEST
KIND FILE ------- (07/22/77 14.06.16.) FEEDBACK TEST

MAXIMUM SIGNAL VALUE (INFINITY) IS 7
MAXIMUM DISTRIBUTION SIZE IS 3000

RUN NUMBER 1
PMIN = 1.0000E-08

NEW = 0. INTER = 1, SAVE = 1
FIRST = 10000, LAST = 10000, TRACE = 2.00000

ANALYSIS DETAILS

OP	TYPE	KIND	SIZE
1	5	1	1
2	5	2	1
3	6	60	3
4	3	30	4
5	6	61	3
6	13	301	9
7	13	603	23
8	10	0	23
9	10	0	23
10	2	0	23
11	13	303	45
12	13	601	75
13	10	0	75
14	10	0	75
15	2	0	75
16	13	305	113
17	10	0	113
18	2	0	113
19	13	701	159
20	2	0	159
21	10	0	159
22	9	900	159
23	13	307	213
24	10	0	213
25	2	0	213
26	7	70	196
27	8	800	196
28	10	0	183
29	2	0	183

TABLE 11.10.3. GO3 OUTPUT DATA.

TABLE 11.10.3 (Continued)

OP	TYPE	KIND	SIZE
30	7	71	192
31	7	72	189
32	3	31	217
33	6	62	209
34	3	32	231
35	1	10	231
36	8	80	231
37	10	0	215
38	2	0	215
39	10	0	215
40	10	0	181
41	2	0	180
42	10	0	148
43	2	0	132
44	10	0	104
45	10	0	104
46	2	0	104
47	2	0	103
48	10	0	94
49	9	900	59
50	10	0	42
51	2	0	37

FINAL EVENT TABLE (INFINITY = 7)

SIGNALS AND THEIR VALUES

PROBABILITY	13	16	19
.0000000410	1	7	7
.0000001857	2	7	7
.0000009569	7	1	1
.0000009753	1	1	1
.0000019138	7	1	0
.0000019138	7	1	4
.0000019138	3	3	0
.0000019138	3	3	1
.0000019176	3	1	7
.0000019331	1	0	7
.0000019369	1	1	4
.0000019564	1	1	0
.0000038276	7	0	0
.0000038276	7	0	4
.0000038352	3	0	7
.0000038506	2	2	4
.0000038545	2	0	7
.0000039010	2	2	0

TABLE 11.10.3 (Continued)

PROBABILITY	13	16	19
.0000048066	2	1	7
.0000048240	2	2	1
.0000057934	2	2	3
.0000058107	1	7	4
.0000115519	2	7	4
.0000174494	2	7	3
.0000292691	7	7	3
.0000401894	3	7	7
.0009839500	7	1	7
.0009849263	7	7	1
.0009850340	1	1	7
.0019195477	3	3	4
.0019755723	7	0	7
.0019795084	7	7	0
.0048675783	2	2	7
.0057873498	3	7	4
.0069850585	7	7	7
.0097203133	7	7	4
.9636445067	3	3	7

TOTAL PROBABILITY = .9999936944
TOTAL ERROR = .0000063056

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	13	16	19
0	0.0000000000	.0019928503	.0019930210
1	.0009976874	.0019853612	.0009935963
2	.0049237955	.0048859474	0.0000000000
3	.9714011740	.9655678819	.0000525119
4	0.0000000000	0.0000000000	.0174561023
7	.0266710375	.0255616537	.9794984629

REFERENCES

1. Apostolakis, G., Computer Oriented Fault Tree Construction, Draft, p.14, UCLA, August 1976.

2. Vesely, W.E., and R.E. Narum, PREP and KITT: Computer Codes for the Automatic Evaluation of a Fault Tree, IN-1349, Idaho Nuclear Corporation.

3. Yellman, Ted W., IEEE Transactions on Reliability, volume R-24, No. 5, p.344, December 1975 Comments on "Fault Trees - A State of the Art Discission".

4. Singh, Jagjit, Great Ideas of Operations Research, LCCCN 68-14995, p.149, Dover Publications, Inc., New York, 1968.

5. Gateley, W.Y., and Williams, R.L., System Operation Simulation Program, Kaman Sciences Corp., K-76-98U(R), Aug. 1976.

6. Reliability and Maintainability Predictions of the ASR-8, General Dynamics, July 1976.

7. Williams, R.L., and W.Y. Gateley, MTTF and MTTUM Analyses of FAA Equipments, Kaman Sciences Corporation, K-76-106u(R), p.3, 21 Sept. 1976.

8. Reliability Systems Engineering Study of the TRACON Critical Power Distribution System, Department of Transportation, Federal Aviation Administration Airway Facilities Directorate, AFA 200B (contractor Illinois Institute of Technology Research Institute), Study No. 1, 15 September 1975.

9. A Procedure to Derive MTBF for Complex Systems Using GO, Kaman Sciences Corporation, K-76-42U(R), 15 March 1976.

10. Reliability Engineering, ARINC Research Corporation, Prentice-Hall, Inc., 1964, pp.235-236.

11. Kunz, Kaiser A., Numerical Analysis, McGraw-Hill Book Company, Inc., 1957, LCCCN56-13395.

12. Curtis, Gale B., A Comparison of Reliability Evaluation Methodologies, Kaman Sciences Corporation, K-74-83U(R), 3 July 1974.

13. Stoddard, Dan W., and Charles W. Hays, A Reliability Analysis of a Typical Electro-Mechanical System, KN-63-72A(M), 26 July 1963.

14. Williams, R.L., and Wilson Y. Gateley, Preliminary Reliability Assessment of the Agglomerating Burner Gasification Process Development Unit, Kaman Sciences Corporation, K-76-131U(R), 15 November 1976.

15. Adams, Corder, Goldberger, Agglomerating Burner Gasification Process, Battelle Columbus Laboratories, FE-1513-38, February 1976.

16. Wood, Donald E., Network Analysis For Reliability Using the GO Methodology, Kaman Sciences Corporation, 29 June 1976.

17. Spryatos, N., B.A. Bowen, J. deMercado, "A Method for Calculation of Network Reliability", IEEE Transactions On Reliability. Volume R-25, June 1976, pp.71-76.

18. Williams, R. L., Reliability Assessment of A Servo Power Supply, Kaman Sciences Corporation, K76-59U(R), 1 May 1976.

19. Williams, R.L., and Wilson Y. Gateley, Analysis of A Diesel Generator Using the GO Methodology, Kaman Sciences Corporation, K77-75U(R), 30 November 1977.

APPENDIX A
PROBABILITY REVIEW

This appendix contains a very brief review of some of the aspects of the theory of probability which are of particular importance in understanding and using GO. Although not intended to replace a formal background of probability, it may provide oil for some of the rusty parts of the possibly long unused machinery which resides in the mind of the reader.

An experiment is an undefined concept which is characterized by one or more outcomes. If certain outcomes always occur, the experiment is deterministic, but if the outcomes can be different from trial to trial of the experiment (due to unexplained vagaries of the experiment), the experiment is non-deterministic (random, stochastic) and is the basic subject matter of probability.

The set of all possible outcomes of an experiment is called the sample space. Each outcome is assigned (in some manner external to probability theory itself) a probability. In most cases the probability of an outcome can be conveniently interpreted as the relative frequency of the occurrence of that outcome during a very large ("infinite") number of trials of the experiment.

An event is some logical (Boolean) combination of outcomes. Two events are mutually exclusive (disjoint) if they cannot occur simultaneously during a single trial of the experiment.

The probability of an event is computed using the basic axioms of probability: let A and B be events and $\bar{A}$ be the complement of A ("not A"); let P(A) be the probability of A; then:

$0 \leq P(A) \leq 1$

$P(\bar{A}) = 1 - P(A)$

$P(A \text{ or } B) = P(A) + P(B)$ if A and B are mutually exclusive.

Numerous theorems follow from the axioms, probably the most useful being

$$P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$$

Events A and B are independent if $P(A \text{ and } B) = P(A)\,P(B)$. Independence is usually postulated by the user on the intuitive basis that the possible occurrences of A and B have no effect on each other.

In many experiments it is desirable to assign a real number to each outcome of an experiment - for example, the number of heads appearing when two coins are tossed simultaneously. Such an assignment defines the values of a random variable (mathematically a random variable is a function whose domain is the sample space and whose range is a subset of the real numbers). Each value of a random variable has a probability which is determined by the probabilities of the experiment outcomes which produce the particular value. This assignment of probabilities to the values of a random variable is called the (probability) distribution of the random variable. If more than one random variable is defined, a joint distribution is obtained.

A random variable is continuous if its values form a continuous set of numbers (frequently an infinite interval such as $(-\infty, \infty)$ or $(0, \infty)$) and discrete if the value set is a discrete set of numbers. (All random variables used in GO are inherently discrete although in some cases they may represent discrete approximations of continuous ones.)

Let A be a discrete random variable. The distribution of A can be characterized by either a probability mass function (pmf) which is defined by $f(x)=P(A=x)$ or a cumulative distribution function (cdf) defined by $F(x)=P(A\leq x)$.

The term "probability density function" (pdf) is sometimes incorrectly used instead of pmf. A pdf is a function related to a continuous random variable and has the basic property that its definite integral from x to y gives the value of $P(x < B \leq y)$ where B is the continuous random variable.

When a joint distribution of several random variables is given (usually defined by means of a joint pmf or pdf), it is possible to find the distribution (or joint distribution) of one (or several) of the random variables by summing or integrating the initial joint distribution over the values of the unwanted variables. The resulting distributions are sometimes referred to as marginal (or joint marginal) distributions from the fact that the marginal distribution of each of two discrete random variables whose joint distribution is defined by means of a two- dimensional table are readily found by summing the rows and columns and writing the results in the table margins.

Two random variables are independent if and only if their joint distribution is the product of the two marginals. More generally, independence of two random variables means that any event involving just one of the variables is independent of any event involving just the other one. This concept - or more properly its generalization to several variables - is used repeatedly in GO, and is the basis of the sequential nature of a GO analysis.

APPENDIX B
OPERATIONAL EXAMPLE

This example has been designed to illustrate many of the operational features of GO. The "problem" is completely artificial and includes most of the regular operator types as well as three supertypes.

Supertype definitions are shown in Figure B.1 and the problem GO chart in Figure B.2. Three supertypes, 300, 301, and 302 have been defined. Type 300 is nested within 301. Type 301 includes two dummy kind numbers, 2000 and 2001; 2001 is passed on to the nested type 300. Figure B.3 shows the entire data deck for the run (the control card deck is not shown because these cards will differ from system to system).

The printouts from the GO run (which includes four sensitivity runs of GO3) are shown in Figure B.4, B.5 and B.6 for GO1, GO2, and GO3 respectively. Some of the pertinent features of each of these are commented upon below.

B.1 GO1

1. The value of VALUES has been set to 10 on the parameter card. Failure to define VALUES (or INFIN) explicitly will always result in an error because the default value of 200 is out of range.
2. The value of BIAS has been set to 4000 on the parameter card. This leads eventually to the creation of signal 4001 by operator 2. The signal number would have been 5001 if the default value of 5000 for BIAS has been used.
3. All supertypes (ST) have been defined at the beginning. The definition of ST 302 could have been deferred until after the super-operator 301

card. Also the definition of ST 300 could have been placed after that of ST 301 even though 300 is nested within 301.

4. When calling supertype 302, we have defined the user-supplied signal bias of 8000, and this resulted in signals 8001 and 8002 being generated by operators 4 and 5 respectively. If we had used the automatic biasing (by writing 0 instead of 8000 on the superoperator card), these signal numbers would have been 4002 and 4003 (recall that 4001 has been generated previously by operator 2).
5. Although we intend signals 413, 9998, and 214 to be final outputs, we have purposely left them off the final signal card to show how they are put on the list. Note, however, if one of these had been used by another operator, it would not have been unused and consequently would not have been put on the final signal list by default.

B.2 GO2

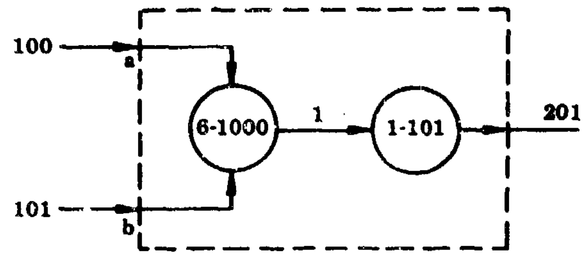
1. The perfect case option has been selected by punching an "X" (a nonblank) in column 80 of the name card. This card is reproduced on the first line, and also the message "PERFECT CASE RUN" is printed.
2. Data for the operators of type 1, 3, 6, and 7 have not been included in the kind deck. The perfect data for these have been generated as kinds 101, 103, 106, and 107 respectively as requested by the operator file (note the minus sign preceding the use frequency in the summary table).
3. Note the format of the data for kind 999. It is a multi-card record, and the data are laid out to facilitate reading by the user.

B.3 GO3

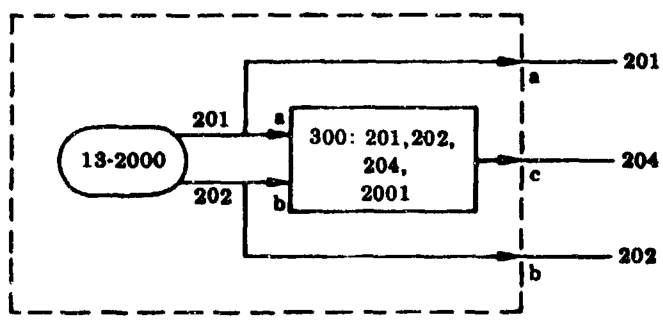
1. The printout shows the results of the regular run (run 1) and four sensitivity runs (runs 2 to 5). In all of the sensitivity runs we have included parameter changes - in fact, runs 2 and 3 involve only parameter changes and no new kind data. This is definitely atypical because in most work there will seldom be any parameter changes in the sensitivity runs. When no parameter changes are to be made, the column 80 character of the sensitivity run name card will be blank, and the parameter card will not be present.
2. In run 1 we have made a full trace of operators 6 and 7 and have used the default values of 1.E-8 for PMIN. We note that the total error is 0 (to 10 decimal places), and consequently there was probably no pruning
3. In run 2 the value of TRACE has been increased from 0.0 to 0.15 and everything else left the same. Note that this has reduced the number of terms printed during the tracing, the missing terms being those whose probabilities are less than TRACE.
4. In run 3 the trace has been removed (by setting FIRST to 10000, and the intermediate details for all operators are requested by setting INTER=1. Also the selective operator kind change is illustrated. The new kind data for kind #101 has the operator number 17 appended to it. This results in this data being applied to only OP. #17 while the other operators using kind #101 (OP. #4) continue to use the kind file data. By comparing the results of this run with those of run 2, the interested reader can verify that this is indeed what happened.

5. In run 4 the value of PMIN is set to 0.05 to demonstrate the effect of pruning - a rather severe effect in this case. The intermediate distribution sizes in this run should be compared with those in run 3 (the kind change in run 3 affected only the final distribution) in order to see how these sizes are decreased. Also note the total error has now increased to 0.2944, an amount which would be intolerable in most real problems.
6. Finally, in run 5, we set PMIN to 0.0 and define new kind data for type 1, 3, 6, and 7. Note that the selective operator option has not been used, and the new kind data for kind #101 will apply to OP #3, #4, and #17. The sizes of the distribution have generally increased as would be expected because of the additional operational modes of the operators affected by the new kind data.
7. The reader should note the fact that in sensitivity runs, changes in parameter values are maintained until explicitly rechanged whereas changes in kind data apply only to the single run in which they were made. (An exception to this is the parameter SAVE which is always reset to the default value of 0 at the end of each run.)

300: 100,101,201,1000



301: 201,202,204,2000,2001



302: 100,200

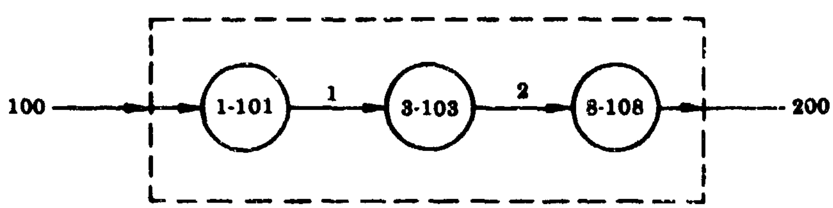


FIGURE B.1. SUPERTYPES.

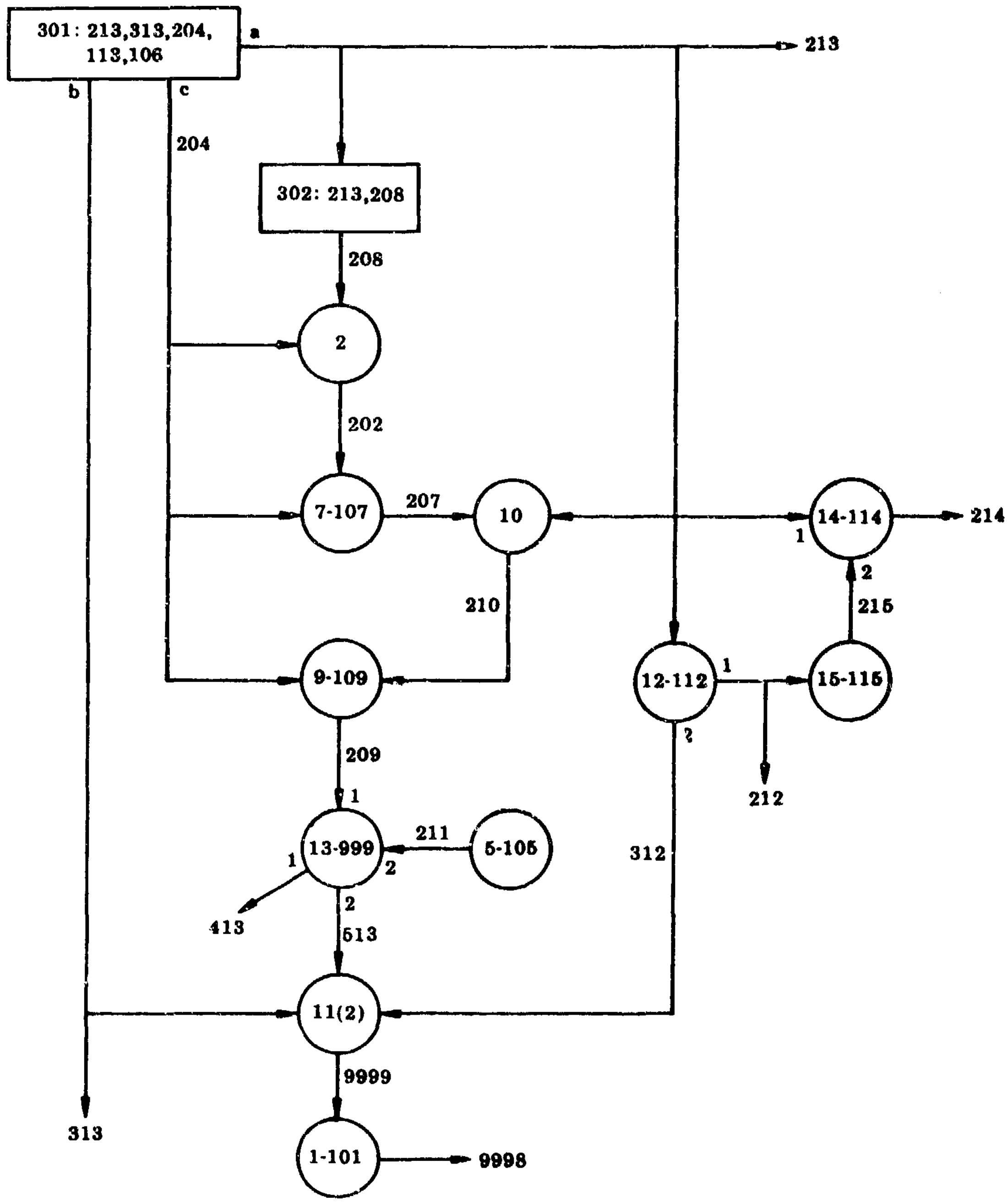


FIGURE B.2. GO CHART.

```
GD USERS MANUAL PROBLEM - OPERATORS
 $PARAM  VALUES=10,  BIAS=4000  $
300  -1  100  101  201  1000  $   THE NESTED SUPERTYPE DEFINITION
6  1000  100  101  1  $
1  101  1  201  $
--- EOR CARD (7/8/9) ---
301  -1  201  202  204  2...  2001  $  FIRST MAIN SUPERTYPE DEFINITION
13  2000  0  2  201  202  $
300  3  201  202  204  2001  $
--- EOR CARD (7/8/9) ---
302  -1  100  200  $  THE SECOND MAIN SUPERTYPE DEFINITION
1  101  100  1  $
3  103  1  2  $
8  108  2  200  $
--- EOR CARD (7/8/9) ---
301  0  213  313  204  113  166  $  FIRST SUPEROP - AUTOMATIC BIAS
302  8000  213  208  $  SECOND SUPEROP - USER-SUPPLIED BIAS
2  0  2  208  204  202  $
7  107  202  204  207  $
10  0  2  207  213  210  $
9, 109, 210,,,,  204, ,209      $ COMMAS CAN ALSO BE USED AS SEPARATORS
5  105  211  $
13  999  2  209  211  2  413  513  $  TYPE 13 WITH INPUTS
12  112  213  2  212  312  $
15  115  212  215  $
14  114  2  213  215  214  $
11  2  3  312  513  313  9999  $
1  101  9999  9998  $
0  212  213  313  $  THE FINAL SIGNAL CARD
--- EOR CARD (7/8/9) ---
GD USERS MANUAL PROBLEM - KINDS                                          X
105 5 2 6 .4 7 .6$
108  8  3  -1  .3  0  .5  2  .2  $
109 9 2 0 1 1 1$
112 12 2 .3 .5$
113 13 0 2 1 2 1 2 .4 3 4 .6$
114  14  2  1  -1  3  $
115  15  -1  4  1  8  .02  1.  $
999 13 2 2 3
    2 6       2     8 9 .4     9 8 .6
    4 7       1     3 0 1.
    2 8       1     8 8 1.  $
--- EOR CARD (7/8/9) ---
REGULAR RUN (RUN 1) - FULL TRACE OF OPS 6 AND 7
 $PARAM  FIRST=6,  LAST=7,  TRACE=0  $
RUN 2 - INCREASE TRACE TO 0.15 FOR PARTIAL TRACE OF SAME OPS            X
 $PARAM  TRACE=0.15  $
--- EOR CARD (7/8/9) ---
RUN 3 - REMOVE TRACE, PRINT INTERMEDIATE DETAILS, MAKE SELECTIVE KIND CHANGE   X
 $PARAM  FIRST=10000,  INTER=1  $
101  1  .5  .5,  17  $
--- EOR CARD (7/8/9) ---
RUN 4 - RAISE PMIN TO 0.05                                              X
 $PARAM  PMIN=0.05  $
--- EOR CARD (7/8/9) ---
RUN 5 - PUT PMIN TO 0, MAKE ALL PERFECT KINDS NON-PERFECT               X
 $PARAM  PMIN=0.0  $
101  1  .8  .2  $
103  3  .7  .2  .1  $
106  6  .7  .2  .1  $
107  7  .7  .2  .1  $
--- EOR CARD (7/8/9) ---
--- EOF CARD (6/7/8/9) ---
```

FIGURE B.3.   DATA DECK.

```
GO USERS MANUAL PROBLEM - OPERATORS
    RUN ON 11/04/76 AT 11.27.15.

VALUES =  10, BIAS = 4000, OPS = 1, SIGNALS = 1, ERRORS = 25


OPERATOR DATA

 OP   DATA
----  ----------------------------------------------------------------
XXXX  300  -1  100  101  201  1000  $   THE NESTED SUPERTYPE DEFINITION
XXXX  6  1000  100  101  1  $
XXXX  1  101  1  201  $
----  END OF SUPER TYPE 300
XXXX  301  -1  201  202  204  2000  2001  $  FIRST MAIN SUPERTYPE DEFINITION
XXXX  13  2000  0  2  201  202  $
XXXX  300  0  201  202  204  2001  $
----  END OF SUPER TYPE 301
XXXX  302  -1  100  200  $  THE SECOND MAIN SUPERTYPE DEFINITION
XXXX  1  101  100  1  $
XXXX  3  103  1  2  $
XXXX  8  108  2  200  $
----  END OF SUPER TYPE 302
$$$$  301  0  213  313  204  113  106  $  FIRST SUPEROP - AUTOMATIC BIAS
   1  (L=1)       13  113    0    2  213  313
$$$$  (L=1)      300    0  213  313  204  106
   2  (L=2)        6  106  213  313 4001
   3  (L=2)        1  101 4001  204
$$$$  302  8000  213  208  $  SECOND SUPEROP - USER-SUPPLIED BIAS
   4  (L=1)        1  101  213 8001
   5  (L=1)        3  103 8001 8002
   6  (L=1)        8  108 8002  208
   7  2  0  2  208  204  202  $
   8  7  107  202  204  207  $
   9  10  0  2  207  213  210  $
  10  9, 109, 210,,,  204, ,209      $ COMMAS CAN ALSO BE USED AS SEPARATORS
  11  5  105  211  $
  12  13  999  2  209  211  2  413  513  $  TYPE 13 WITH INPUTS
  13  12  112  213  2  212  312  $
  14  15  115  212  215  $
  15  14  114  2  213  215  214  $
  16  11  2  3  312  513  313  9999  $
  17  1  101  9999  9998  $
////  0  212  213  313  $  THE FINAL SIGNAL CARD


SIGNAL DATA

        SOURCE OPER.
SIGNAL  NUM TYPE KIND  USING OPERATORS (- IF DELETED AT)
------  ---- ---- ----  ----------------------------------------------
  202     7   2     0    -8
  204     3   1   101     7     8   -10
  207     8   7   107    -9
  208     6   8   108    -7
  209    10   9   109   -12
  210     9  10     0   -10
  211    11   5   105   -12
  212    13  12   112    14
  213     1  13   113     2     4     9    13    15
  214    15  14   114
  215    14  15   115   -15
  312    13  12   112   -16
  313     1  13   113     2    16
  413    12  13   999
  513    12  13   999   -16
 4001     2   6   106    -3
 8001     4   1   101    -5
 8002     5   3   103    -6
 9998    17   1   101
 9999    16  11     2   -17

****  SIGNAL  413 NOT USED, PUT IN FINAL SIGNAL LIST.

****  SIGNAL 9998 NOT USED, PUT IN FINAL SIGNAL LIST.

****  SIGNAL  214 NOT USED, PUT IN FINAL SIGNAL LIST.

NUMBER OF OPERATORS     =   17
NUMBER OF SIGNALS       =   20
MAX NUMBER ACTIVE       =    7
MAX SIGNAL LIST SIZE    =    7
NUMBER OF SIGNAL VALUES=   10
NUMBER OF SIGNALS/WORD =   15
NUMBER OF WORDS/TERM    =    1

FINAL SIGNALS =   212  213  313  413 9998  214
```

FIGURE B.4.   GO1 RUN.

```
GO USERS MANUAL PROBLEM - KINDS                                        X
    RUN ON 11/04/76 AT 11.29.16.

OPERATOR FILE --- ( 11/04/76  11.27.15.) GO USERS MANUAL PROBLEM - OPERATORS

--- PERFECT CASE RUN ---

RECORD KIND DATA
------ ----------------------------------------------------------------
     1  105 5 2 6 .4 7 .6$
     2  108  8  3  -1  .3  8  .5  2  .7  $
     3  109 9 2 8 1 1 1$
     4  112 12 2 .3 .5$
     5  113 13 0 2 1 2 1 2 .4 3 4 .6$
     6  114  14  2  1  -1  3  $
     7  115  15  -1  4  1  8  .62  1.  $
     8  999 13 2 2 3
            2 6        2     8 9 .4     9 8 .6
            4 7        1     5 8 1.
            2 8        1     8 8 1.  $
--- ----------------------------------------------------------------


USE SUMMARY TABLE. ENTRY = KIND/TYPE(FREQUENCY)
   (FREQUENCY IS NEGATIVE FOR PERFECT KINDS.)

101/ 1(   -3)  103/ 3(   -1)  105/ 5(    1)  106/ 6(   -1)  107/ 7(   -1)  108/ 8(    1)  109/ 9(    1)  112/12(    1)  113/13(    1)
114/14(    1)  115/15(    1)  999/13(    1)

NUMBER OF KINDS INPUT-----   8
NUMBER USED - NONPERFECT--   6
NUMBER USED - PERFECT-----   4
```

FIGURE B.5.   GO2 RUN.

REGULAR RUN (RUN 1) - FULL TRACE OF OPS 6 AND 7
RUN ON 11/04/76 AT 11.31.05.

OPERATOR FILE --- (11/04/76 11.27.15.) GO USERS MANUAL PROBLEM - OPERATORS
KIND FILE ------- (11/04/76 11.29.16.) GO USERS MANUAL PROBLEM - KINDS

MAXIMUM SIGNAL VALUE (INFINITY) IS 9
MAXIMUM DISTRIBUTION SIZE IS 3000

RUN NUMBER 1
PMIN =1.0000E-08
NEW = 0, INTER = 0, SAVE = 0
FIRST = 6, LAST = 7, TRACE = 0.

ANALYSIS DETAILS

OP	TYPE	KIND	SIZE						
6	8	108	6						
.1200000000	213(	1)	313(	2)	204(	2)	208(	0)
.2000000000	213(	1)	313(	2)	204(	2)	208(	1)
.0800000000	213(	1)	313(	2)	204(	2)	208(	3)
.1800000000	213(	3)	313(	4)	204(	4)	208(	2)
.3000000000	213(	3)	313(	4)	204(	4)	208(	3)
.1200000000	213(	3)	313(	4)	204(	4)	208(	5)
7	2	0	6						
.1200000000	213(	1)	313(	2)	204(	2)	202(	0)
.2000000000	213(	1)	313(	2)	204(	2)	202(	1)
.0800000000	213(	1)	313(	2)	204(	2)	202(	2)
.1800000000	213(	3)	313(	4)	204(	4)	202(	2)
.3000000000	213(	3)	313(	4)	204(	4)	202(	3)
.1200000000	213(	3)	313(	4)	204(	4)	202(	4)

FINAL EVENT TABLE (INFINITY = 9)

	SIGNALS AND THEIR VALUES					
PROBABILITY	212	213	313	413	9996	214
.0132400000	9	1	2	8	9	0
.0153600000	9	1	2	9	8	0
.0153600000	1	1	2	8	9	0
.0230400000	1	1	2	9	8	3
.0256000000	9	1	2	8	2	0
.0544000000	9	1	2	9	9	0
.0576000000	9	3	4	5	4	2
.0624000000	9	3	4	9	9	2
.0816000000	1	1	2	9	9	3
.0864000000	1	3	4	5	4	5
.0936000000	1	3	4	9	8	5
.1440000000	9	3	4	5	1	2
.1560000000	9	3	4	9	4	2
.1744000000	9	1	2	9	2	0

TOTAL PROBABILITY = 1.0000000000
TOTAL ERROR = .0000000000

FIGURE B.6. GO3 RUN.

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	212	213	313	413	9998	214
0	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	.2953600000
1	.3000000000	.4000000000	0.0000000000	0.0000000000	.1440000000	0.0000000000
2	0.0000000000	0.0000000000	.4000000000	0.0000000000	.2000000000	.4200000000
3	0.0000000000	.6000000000	0.0000000000	0.0000000000	0.0000000000	.1046400000
4	0.0000000000	0.0000000000	.6000000000	0.0000000000	.3000000000	0.0000000000
5	0.0000000000	0.0000000000	0.0000000000	.2880000000	0.0000000000	.1800000000
8	0.0000000000	0.0000000000	0.0000000000	.0512000000	.0384000000	0.0000000000
9	.7000000000	0.0000000000	0.0000000000	.6608000000	.3176000000	0.0000000000

****

NEW KIND DATA --- RUN 2 - INCREASE TRACE TO 0.15 FOR PARTIAL TRACE OF SAME OPS X

RECORD KIND DATA

RUN NUMBER 2
PMIN =1.0000E-08
NEW = 0, INTER = 0, SAVE = 0
FIRST = 6, LAST = 7, TRACE = .150000

ANALYSIS DETAILS

OP	TYPE	KIND	SIZE						
6	8	106	6						
.2000000000	213(	1)	313(	2)	204(	2)	208(	1)	
.1000000000	213(	3)	313(	4)	204(	4)	206(	2)	
.3000000000	213(	3)	313(	4)	204(	4)	208(	3)	
7	2	8	6						
.2000000000	213(	1)	313(	2)	204(	2)	202(	1)	
.1000000000	213(	3)	313(	4)	204(	4)	202(	2)	
.3000000000	213(	3)	313(	4)	204(	4)	202(	3)	

FINAL EVENT TABLE (INFINITY = 9)

SIGNALS AND THEIR VALUES

PROBABILITY	212	213	313	413	9998	214
.0102400000	9	1	2	8	9	0
.0153600000	9	1	2	9	8	0
.0153600000	1	1	2	8	9	0
.0230400000	1	1	2	9	8	3
.0256000000	9	1	2	8	2	0
.0544000000	9	1	2	9	9	0
.0576000000	9	3	4	5	4	2
.0624000000	9	3	4	9	9	2
.3016000000	1	1	2	9	9	3
.0864000000	1	3	4	5	4	5
[illegible]	1	3	4	9	9	5
.1440000000	9	3	4	5	1	2
.1960000000	9	3	4	9	4	2
.1764000000	9	1	2	9	2	0

FIGURE B.6 (Continued). GO3 RUN.

BEST AVAILABLE COPY

TOTAL PROBABILITY = 1.0000000000
TOTAL ERROR = .0000000000

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	212	213	313	413	9996	214
0	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	.2953600000
1	.3000000000	.4000000000	0.0000000000	0.0000000000	.1440000000	0.0000000000
2	0.0000000000	0.0000000000	.4000000000	0.0000000000	.2000000000	.4200000000
3	0.0000000000	.6000000000	0.0000000000	0.0000000000	0.0000000000	.1046400000
4	0.0000000000	0.0000000000	.6000000000	0.0000000000	.3000000000	0.0000000000
5	0.0000000000	0.0000000000	0.0000000000	.2880000000	0.0000000000	.1800000000
8	0.0000000000	0.0000000000	0.0000000000	.0512000000	.0384000000	0.0000000000
9	.7000000000	0.0000000000	0.0000000000	.6608000000	.3176000000	0.0000000003

****

NEW KIND DATA --- RUN 3 - REMOVE TRACE, PRINT INTERMEDIATE DETAILS, MAKE SELECTIVE KIND CHANGE X

RECORD KIND DATA

1 101 1 .5 .5, 17 5

RUN NUMBER 3
PMIN =1.0000E-08
NEW = 0, INTER = 1, SAVE = 0
FIRST = 10000, LAST = 7, TRACE = .153000

ANALYSIS DETAILS

OP	TYPE	KIND	SIZE
1	13	113	2
2	6	106	2
3	1	101	2
4	1	131	2
5	3	103	2
6	6	108	6
7	2	0	6
8	7	107	6
9	10	0	4
10	9	109	4
11	5	105	8
12	13	999	5
13	12	112	15
14	15	115	15
15	14	114	15
16	11	-2	14
17	1	101	16

FINAL EVENT TABLE (INFINITY = 9)

PROBABILITY	SIGNALS AND THEIR VALUES 212	213	313	413	9996	214
.0076800000	9	1	2	9	8	0
.0115200000	1	1	2	9	8	3
.0128000000	9	1	2	8	2	0
.0153600000	1	1	2	8	9	0

FIGURE B.6. (Continued). GO3 RUN.

.0230400000	9	1	2	8	9	0
.0288000000	9	3	4	5	4	2
.0432000000	1	3	4	5	9	5
.0432000000	1	3	4	5	4	5
.0720000000	9	3	4	5	1	2
.0780000000	9	3	4	9	4	2
.0872000000	9	1	2	8	2	0
.0931200000	1	1	2	9	9	3
.0936000000	1	3	4	9	9	5
.1008000000	9	3	4	5	9	2
.1404000000	9	3	4	9	9	2
.1492800000	9	1	2	9	9	0

TOTAL PROBABILITY = 1.0000000000
TOTAL ERROR = .0000000000

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	212	213	313	413	9998	214
0	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	.2953600000
1	.3000000000	.4000000000	0.0000000000	0.0000000000	.0720000000	0.0000000000
2	0.0000000000	0.0000000000	.4000000000	0.0000000000	.1000000000	.4200000000
3	0.0000000000	.6000000000	0.0000000000	0.0000000000	0.0000000000	.1046400000
4	0.0000000000	0.0000000000	.6000000000	0.0000000000	.1500000000	0.0000000000
5	0.0000000000	0.0000000000	0.0000000000	.2880000000	0.0000000000	.1800000000
8	0.0000000000	0.0000000000	0.0000000000	.0512000000	.0192000000	0.0000000000
9	.7000000000	0.0000000000	0.0000000000	.6600000000	.6588000000	0.0000000000

NEW KIND DATA --- RUN 4 - RAISE PMIN TO 0.05

RECORD KIND DATA

RUN NUMBER 4
PMIN =5.0000E-02
NEW = 0, INTER = 1, SAVE = 0
FIRST = 10000, LAST = 7, TRACE = .150000

ANALYSIS DETAILS

OP	TYPE	KIND	SIZE
1	13	113	2
2	6	106	2
3	1	101	2
4	1	101	2
5	3	103	2
6	8	108	8
7	2	0	6
8	7	107	6
9	10	0	4
10	9	109	4
11	5	105	5
12	13	999	5
13	12	112	8
14	15	115	8
15	14	114	8
16	11	-2	8

FIGURE B.6. (Continued). GO3 RUN.

17 1 101 8

FINAL EVENT TABLE (INFINITY = 9)

PROBABILITY	212	213	313	413	9998	214
	SIGNALS AND THEIR VALUES					
.0528000000	9	3	4	9	9	2
.0576000000	9	3	4	5	4	2
.0576000000	1	1	2	9	9	3
.0792000000	1	3	4	9	9	5
.0864000000	1	3	4	5	4	5
.0960000000	9	1	2	9	2	0
.1320000000	9	3	4	9	4	2
.1440000000	9	3	4	5	1	2

TOTAL PROBABILITY = .7056000000
TOTAL ERROR = .2944000000

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	212	213	313	413	9998	214
0	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	.0960000000
1	.2232000000	.1536000000	0.0000000000	0.0000000000	.1440000000	0.0000000000
2	0.0000000000	0.0000000000	.1536000000	0.0000000000	.0960000000	.3864000000
3	0.0000000000	.5520000000	0.0000000000	0.0000000000	0.0000000000	.0576000000
4	0.0000000000	0.0000000000	.5520000000	0.0000000000	.2760000000	0.0000000000
5	0.0000000000	0.0000000000	0.0000000000	.2880000000	0.0000000000	.1656000000
9	.4824000000	0.0000000000	0.0000000000	.4176000000	.1896000000	0.0000000000

NEW KIND DATA --- RUN 5 - PUT PMIN TO 0, MAKE ALL PERFECT KINDS NON-PERFECT X

RECORD KIND DATA

1	101	1	.8	.2	$	
2	103	3	.7	.2	.1	$
3	106	6	.7	.2	.1	$
4	107	7	.7	.2	.1	$

RUN NUMBER 5
PMIN =0.
NEW = 0, INTER = 1, SAVE = 0
FIRST = 10000, LAST = 7, TRACE = .150000

ANALYSIS DETAILS

OP	TYPE	KIND	SIZE
1	13	113	2
2	6	106	6
3	1	101	5
4	1	101	12
5	3	103	16
6	8	108	30
7	2	0	22
8	7	107	26

FIGURE B.6. (Continued). GO3 RUN.

9	10	0	17
10	9	109	6
11	5	105	12
12	13	999	5
13	12	112	15
14	15	115	15
15	14	114	15
16	11	-2	14
17	1	101	17

FINAL EVENT TABLE (INFINITY = 9)

SIGNALS AND THEIR VALUES

PROBABILITY	212	213	313	413	9998	214
.0019475251	1	1	2	9	9	0
.0051934003	9	1	2	9	8	0
.0064917504	9	1	2	8	9	0
.0064917504	1	1	2	8	9	0
.0075209472	1	3	4	5	9	5
.0077901005	1	1	2	9	8	0
.0086550672	9	1	2	8	2	0
.0175488768	9	3	4	5	9	2
.0209558592	9	3	4	5	4	2
.0300837888	1	3	4	5	4	5
.0531396480	9	3	4	5	1	2
.1037706240	1	1	2	9	9	3
.1083148493	9	1	2	9	9	0
.1423952648	9	3	4	9	9	2
.1423952640	1	3	4	9	9	5
.1513443328	9	1	2	9	2	0
.1898603520	9	3	4	9	4	2

TOTAL PROBABILITY = 1.0000000000
TOTAL ERROR = .0000000000

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.	212	213	313	413	9998	214
0	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	.2962293760
1	.3000000000	.4000000000	0.0000000000	0.0000000000	.0501396480	0.0000000000
2	0.0000000000	0.0000000000	.4000000000	0.0000000000	.1600000000	.4200000000
3	0.0000000000	.6000000000	0.0000000000	0.0000000000	0.0000000000	.1037706240
4	0.0000000000	0.0000000000	.6000000000	0.0000000000	.2400000000	0.0000000000
5	0.0000000000	0.0000000000	0.0000000000	.1253491200	0.0000000000	.1600000000
8	0.0000000000	0.0000000000	0.0000000000	.0216391680	.0129835008	0.0000000000
9	.7000000000	0.0000000000	0.0000000000	.8530117120	.5368768512	0.0000000000

FIGURE B.6. (Continued). GO3 RUN.

BEST AVAILABLE COPY

The printout in Figure B.8 shows the results of modifying the operator deck of the example in such a manner that numerous data errors are present. Figure B.7 shows the modified operator deck.

The errors and their results are for the most part self-explanatory. Note that the parameter VALUES has not been defined on the parameter card, and consequently the default value of 200 is used. We have also set SIGNALS = 0 to avoid printing the signal table; even with previously occurring errors this table is normally printed (along with the message that it is incomplete) because some useful information may possibly be obtained from it.

Note that the operator numbering is occasionally erratic, and consequently when an error occurs, the numbers assigned to subsequent operators may differ from what they will be after the error is corrected.

Certain errors or combinations of errors may lead to fallacious error messages later on. Consequently, if certain messages appear which seem to have no basis, correct the obvious errors and try again.

Also keep in mind that in general only one error per operator is detected. Thus, when correcting an error, be sure to check the entire card for other errors which went undetected the first time around.

```
GO USERS MANUAL PROBLEM - OPERATORS WITH ERRORS
 $PARAM                    BIAS=4000,  SIGNALS=0  $
300  -1  100  101  201  1000  $
6  1000  100  101  1  $
1  101  1  201  $
--- EOR CARD (7/8/9) ---
301  -1  201  202  204  2000  2001  $
13  2222  0  2  201  202  $ THE 2222 IS A BAD KIND
300  0  201  202  204  2001  $
--- EOR CARD (7/8/9) ---
302  -1  100  200  $
1  101  100  1  $
3  103  1  2  $
8  108  2  203  $
--- EOR CARD (7/8/9) ---
301  0  213  313  204  113  103  $
322  0000  213  208  $  UNDEFINED SUPERTYPE NUMBER
22 0  2  208  204  202  $   BAD TYPE
7  107  202  204  207  $
10  0  2  207  213  210  210  $  THE 2ND 210 IS EXTRA DATA
9, 109, 210,,,  204, ,209      $
5  1050  211  $  1050 IS A BAD KIND
13  999  2  209  211  2  413  513  $
12  112  213  2  212  312  $
15  115  212  -215  $  A BAD SIGNAL NUMBER
14  114  2  213  215  214  $
11  2  3  312  513  313  214  $  214 HAS ALSO AN OUTPUT FOR PREVIOUS OP
1  601  99999 9998  $ 99999 HAS TOO MANY CHARACTERS
6  312  213  313  444  $  FINAL SIGNAL CARD.  444 HAS NOT BEEN ENTERED
--- EOR CARD (7/8/9) ---
```

FIGURE B.7.   GO1 DATA DECK.

BEST AVAILABLE COPY

```
GO USERS MANUAL PROBLEM - OPERATORS WITH ERRORS
    RUN ON 11/04/76 AT 12.40.57.

VALUES = 200, BIAS = 4000, OPS = 1, SIGNALS = 0, ERRORS = 25

**** VALUES IS OUT OF RANGE

------ERROR------

OPERATOR DATA

 OP   DATA
----  -----------------------------------------------------------
XXXX  300  -1  100  101  201  1000  $
XXXX  6  1000  100  101  1  $
XXXX  1  101  1  201  $
----  END OF SUPER TYPE 300
XXXX  301  -1  201  202  204  2000  2001  $
XXXX  13  2222  0  2  201  202  $ THE 2222 IS A BAD KIND
**** KIND  2222 NOT IN LIST. REPLACED BY 999

------ERROR------

XXXX  300  0  201  202  204  2001  $
----  END OF SUPER TYPE 301
XXXX  302  -1  100  200  $
XXXX  1  101  100  1  $
XXXX  3  103  1  2  $
XXXX  8  108  2  200  $
----  END OF SUPER TYPE 302
$$$$  301  0  213  313  204  113  106  $
   1  (L=1)     13  999   0    2  213  313
$$$$  (L=1)    300    0  213  313  204  106
   2  (L=2)      6  106  213  313 4001
   3  (L=2)      1  101 4001  204
$$$$  322  8000  213  208  $  UNDEFINED SUPERTYPE NUMBER
**** UNDEFINED SUPER TYPE

------ERROR------

   4  22  0  2  208  204  202  $   BAD TYPE
**** BAD PARAMETER

------ERROR------

   4  7  107  202  204  207  $
**** INPUT SIGNAL  202 HAS NOT BEEN ENTERED

------ERROR------

   5  10  0  2  207  213  210  210  $  THE 2ND 210 IS EXTRA DATA
**** WRONG AMOUNT OF DATA

------ERROR------

   6  9, 109, 210,,,  204, ,209     $
   7  5  1050 211  $  1050 IS A BAD KIND
**** KIND OUT OF RANGE

------ERROR------

   8  13  999  2  209  211  2  413  513  $
   9  12  112  213  2  212  312  $
```

FIGURE B.8.   GO1 RUN.

```
  10  15  115  212  -215  $  A BAD SIGNAL NUMBER
**** SIGNAL -215 IS OUT OF RANGE

------ERROR------

  11  14  114  2  213  215  214  $
**** INPUT SIGNAL  215 HAS NOT BEEN ENTERED

------ERROR------

  12  11  2  3  312  513  313  214  $  214 HAS ALSO AN OUTPUT FOR PREVIOUS OP
**** SIGNAL  214 REUSED

------ERROR------

  13  1  101  99999 9998  $ 99999 HAS TOO MANY CHARACTERS
**** DATA ITEM EXCEEDS  4 CHARACTERS

------ERROR------

//23  0  212  213  313  444  $  FINAL SIGNAL CARD.  444 HAS NOT BEEN ENTERED

****  FINAL SIGNAL  444 WAS NEVER ENTERED

------ERROR------

SUICIDE BECAUSE OF ERRORS
```

FIGURE B.8. (Continued). GO1 RUN.

PRECEDING PAGE BLANK-NOT FILMED

APPENDIX C
RELIABILITY ANALYSES PERFORMED USING THE GO METHODOLOGY

Reliability Analysis of An HTGR SCRAM System Including Human Interfaces, KSC-1037-1, March 1975. Study performed for the Division of Reactor Research and Development, U.S. Energy Research and Development Administration.

"GO" Evaluation of A Pressurized Water Reactor Spray System, EPRI 350-1, August 1975, KSC.

A Procedure To Derive MTBF For Complex Systems Using GO, K-76-42U(R), 15 March 1976, KSC. R. Larry Williams, Wilson Y. Gateley. Study performed in conjunction with the Airway Facilities Directorate of the Federal Aviation Administration.

Joint Army/ERDA SPARTAN Warhead Section Safety and Reliability Analysis (U) , RCS: SARQA 116, 30 September 1975, Picatinny Arsenal, Dover, New Jersey, SRD.

Joint Army/ERDA SPRINT Warhead Section Safety and Reliability Analysis (U), RCS: SARQA 116, 15 October 1975, Picatinny Arsenal, Dover, New Jersey, SRD.

Special Safety Study-Quantitative Assessment of Premature C4 Motor Ignition and Consequences (U), K-75-394(R), 14 November 1975, SRD. R. Larry Williams. Study performed for the Strategic Systems Project Office of the U.S. Navy.

Safety and Reliability Analysis of the XM134E1 Arm Safe Device. SARPA-QA-N-29-75, 12 November 1975, Picatinny Arsenal. Gale B. Curtis.

Effectiveness Evaluation of the 155MM Projectile M549, HE, RA, K-75-79U(R), 12 September 1975.

Demonstration of A Reliability Assessment Methodology As Applied To Nuclear Power Plant Systems, K-74-30U(R), Noel J. Becar, R. Larry Williams, 9 April 1974, Kaman Sciences Corp.

Joint Army/AEC SPARTAN Warhead Section Reliability and Safety Analysis (U), Picatinny Arsenal 71-9001007, SRD, 1971.

Report of the Abnormal Environments Task Group for the Prelaunch Regime to the SPARTAN Safety and Reliability Working Group (U), Livermore Radiation Laboratory, CMAA-71-151, SRD, 1971.

Joint Army/AEC Reliability and Safety Analysis of the SPRINT Warhead Section (U), Revision A, Picatinny Arsenal, SMUPA-NX-3-71, SRD, 1971. Dan W. Stoddard, . Larry Williams.

Joint Army/AEC SPARTAN Warhead Section Reliability and Safety Analysis, Picatinny Arsenal, SMUPA-QA-N-1-73, SRD, February 1973. Dan W. Stoddard, R. Larry Williams.

SPARTAN XM55 Interconnecting Box Safety and Reliability Assessment Report (U), K-73-606(R), September 1973, Dan W. Stoddard.

Safety and Reliability Study of the Nike-Zeus Adaptation Kit (U), KN-131-61-20(FR), Dan W. Stoddard, R. Larry Williams.

SPARTAN Warhead Section Response To Postulated In-Cell Accidents (U), Final Report, K-73-689(R), R. Larry Williams, Kaman Sciences Corporation.

Quantitative Assessments of the SPRINT Warhead Section Response To Postulated In-Cell Accidents (U), K-74-205(R), May 1974, R. Larry Williams, Kaman Sciences Corporation.

SPARTAN Warhead Section Integrated Math Model (U), K-74-428, 1 November 1974, Dan W. Stoddard, Kaman Sciences Corporation.

Reliability Analysis of the M1140 Altitude Fuze (U), K-74-440, 7 November 1974, R. Larry Williams, Kaman Sciences Corporation.

Safety and Reliability Analysis of the XM730 Fuze (U), K-75-160(R), 17 April 1975, Sale B. Curtis, Kaman Sciences Corp.

Availability Analysis of the Freon Absorption System for Treating Effluents from Reprocessors, KSC-6212-1, Donald E. Wood, 16 August 1976.

A Standardized Munitions Effectiveness Methodology, K-75-57U(R), 15 October 1975, R. Larry Williams, W. Tierce Long, Walter Windholtz.

Reliability Assessment of A Servo Power Supply, K-76-59U(R), 1 May 1976, R. Larry Williams.

A Comparison of Reliability Evaluation Methodologies, K-74-83U(R), 3 July 1974, Gale B. Curtis.

Results of Demonstration of A Reliability Assessment Methodology Using Two Coal Conversion Plant Models, Contract No. 14-32-001-1788, Office of Assistant Administrator - Fossil Energy, ERDA, 6 June 1975.

Kaman Input to NASA/ECON Low Cost Payloads Study, Noel J. Becar, 23 August 1976, Kaman Sciences Corporation.

Safety Analysis of An HTGR Plant Including Emergency Shutdown, K76-124U(R), Noel J. Becar, Donald E. Wood, 30 September 1976, Kaman Sciences Corporation (2 Volumes)

Analysis of Effect of Human Interactions on HTGR Safety and Reliability, KSC-1037-2, Noel J. Becar, 20 June 1975, Kaman Sciences Corporation.

A Comparative Analysis of A PWR SCRAM System, KSC-77-7268-1, Donald E. Wood, Noel J. Becar, Spring 1977, Kaman Sciences Corporation.

A Reliability Assessment Method Including Human Factors, paper presented at 1975 Annual Meeting American Nuclear Society, New Orleans, 9 June 1975, Donald E. Emon (RRD/ERDA), Noel J. Becar (KSC).

Network Analysis for Reliability Using The GO Methodology, Donald E. Wood, 29 June 1976, Kaman Sciences Corporation.

Let's "GO" IEEE -352, Donald E. Wood, 12 July 1976, Kaman Sciences Corporation.

APPENDIX D
GLOSSARY

Each of the technical terms used in GO and certain other terms have been included in this alphabetic glossary along with a brief explanation and/or a reference to the place in the manual where the term is defined or discussed.

Active Signal: A signal which is included in the current probability distribution.

Algorithm: A computational procedure.

Analysis Deck: The card input data for GO3.

Array: A distribution.

BIAS: A GO1 parameter whose value is the initial bias for internal signal numbers in a superoperator.

Deleted Signal: A signal which is deleted or dropped at some operator by effectively summing over its values. The resulting distribution is the joint marginal distribution of all of the active signals except the deleted one.

Distribution: The probability mass function of one or more signals. The adjective "joint" may be used if more than one signal is involved and "marginal" if the distribution has resulted from deleting one or more signals. Also called an "array".

EOR: End-of-record; a card with a multiple 7-8-9 punch in column 1.

ERRORS: A GO1 parameter specifying the number of errors allowed before aborting the GO1 execution.

Final Distribution: The probability distribution which is printed by GO3. The signals (random variables) which are included in this distribution are designated by the user and normally are those whose values are of

	importance in describing the overall operation of the modeled system.
Final Signal:	A signal which is included in the final probability distribution produced by GO.
FIRST:	A GO3 parameter giving the first operator for which tracing is to be performed.
INFIN:	A GO1 parameter; the largest possible signal value; = VALUES- 1.
Infinity:	The largest possible value of a signal; defined as a program parameter by the user.
Input Signal:	A signal whose values must be known by an operator for determining the values of its output signal(s).
INTER:	A GO3 parameter to control the printing of intermediate distribution summaries.
Kind:	Refers to the set of parameters (usually probabilities and signal values) which, together with its type designation, completely defines an operator. Each type has a particular set of required kind data.
Kind Deck:	The card input data for GO2.
LAST:	A GO3 parameter giving the last operator for which tracing is to be performed.
Never:	= infinity.
NEW:	A GO3 parameter used to determine whether the first GO3 run should be a regular or a sensitivity run.

Operator Deck:	The card input data for G01.
OPS:	A G01 parameter to control the printing of the operator data.
Output Signal:	A signal created by an operator. The operator actually creates a new joint probability distribution which includes the output signal.
Parameter Card:	A data card which specifies the user-declared values of certain program parameters.
PMIN:	A G03 parameter specifying the term probability pruning value.
Program Parameter:	A parameter which controls the operation of a GO program. All program parameters have default values which may be altered by the user on the appropriate parameter card.
Regular run:	A G03 run in which the kind file is used as the source for all kind data.
Regular Type:	One of the sixteen (numbers 1 to 3 and 5 to 17) programmed operator types.
SAVE:	A G03 parameter to control the creation of a file containing all distributions created by G03.
Sensitivity Run:	An execution of G03 in which the kind file data for one or more kinds are changed.
Signal:	A random variable created by an operator. The signal values and their probabilities are determined by the operator type, the associated kind data, and the values of the operator input signals, if any.

Signal Value: One of the possible integer values (between zero and †"infinity") which can be assumed by a signal.

SIGNALS: A GO1 parameter to control the printing of the signal cross-reference table.

Super-operator: An operator whose type is a supertype.

Supertype: A user-defined operator type consisting of one or more regular types (or other supertypes) connected together in some operationally meaningful manner.

Time Point: = signal value.

TRACE: A GO3 parameter giving the cut-off probability value for tracing (terms with probabilities less than TRACE are not printed).

Tracing: The printing in GO3 of some terms of the distributions created by some operators; controlled by the parameters TRACE, FIRST, and LAST.

Type: An operator classification. Each type is represented in the GO programs by an algorithm which defines its operational nature. See "regular type" and "supertype".

VALUES: A GO1 parameter; the number of possible signal values; = INFIN + 1.

APPENDIX E
SUMMARIES

E1. Operator and Kind Data Summary

Standard Symbols:

S or S_i:	input signal number.
R or R_i:	output signal number.
VS_i,VR_i:	value of signal S_i, R_i.
K:	kind identification number.
P_g:	probability that operator is good.
P_f:	probability that operator fails (duds).
P_p:	probability that operator prematures.
P_i:	probability that operator is in state i.
:	infinity, the largest possible signal value.

Note: (1) the data on a card must be separated by one or more blanks and/or commas.

(2) Each data record (usually one card) must be terminated with a dollar sign ($).

(3) Multiple cards may be used for one record.

(4) Consult Chapter 6 for nonstandard symbols and additional details.

TYPE	NAME	OPERATOR DATA RECORD	KIND DATA RECORD
1	Two state component	1, K, S, R, \$	K, 1, P_g, P_f \$
2	OR Gate	2, 0, n, $S_1, \ldots, S_n$, R \$	None
3	Triggered Generator	3, K, S, R, \$	K, 3, P_g, P_f, P_p \$
5	Signal Generator	5, K, R, \$	K, 5, n, V_1, $P_1, \ldots, V_n$, P_n \$
6	Normally open contact	6, K, S_1, S_2, R \$	K, 6, P_g, P_f, P_p \$
7	Normally closed contact	7, K, S_1, S_2, R \$	K, 7, P_g, P_f, P_p, \$
8	Increment Generator	8, K, S, R \$	K, 8, n, D_1, $P_1, \ldots,$ D_n, P_n \$
9	Function Operator	9, K, S_1, S_2, R \$	K, 9, n, X_1, $Y_1, \ldots,$ X_n, Y_n \$
10	AND Gate	10, 0, n, $S_1, \ldots, S_n$, R \$	None
11	m-out-of-n Gate	11, m, n, $S_1, \ldots, S_n$, R \$	None
12	Path Splitter	12, K, S, m, $R_1, \ldots,$ R_m \$	K, 12, m, $P_1, \ldots, P_m$ \$
13	Multiple I/O Operator	13, K, n, $S_1, \ldots, S_n$, m, $R_1, \ldots,$ R_m \$	See Chapter 6
14	Linear Combination Generator	14, K, n, $S_1, \ldots, S_n$, R \$	K, 14, n, $a_1, \ldots, a_n$, a_o \$
15	Valve/Probability Gate	15, K, S, R \$	K, 15, V_1, V_2, V_3, V_4, P_1, P_2 \$
16	Actuated Normally Open Gate	16, K, S_1, S_2, R \$	K, 16, P_q, P_f, P_p \$
17	Actuated Normally Closed Contact	17, K, S_1, S_2, R \$	K, 17, P_g, P_f, P_p \$
N	User defined supertype	N,0,$A_1, \ldots, A_m$ \$ (Note: the arguments A_i must correspond with those in the supertype definition).	None

E2. Supertype Summary

A supertype definition subdeck for each user-defined supertype is included in the GO1 operator subdeck and must precede any use of the particular supertype. The subdeck contains

1. Declaration card containing: N, -1, $A_1,\ldots,A_m$ \$
   where N = supertype number (≥ 100)
   m = number of arguments, $m \leq 25$
   A_i = dummy arguments (in any order) may include:
   Input signal numbers (100-199)
   Output signal numbers (200-201)
   Kind numbers (≥ 1000)

2. Operator data cards defining the supertype (note: internal signal numbers not included in the argument list must lie in the range from 1 to 100).
3. An EOR card.

E3 Data Deck Summary

Card Format Styles:

1: 80 column alphanumeric
1X: 79 column alphanumeric, optional control character in column 80
2: Fortran NAMELIST format: \$PARAM ... \$
3: Format-free for operator and kind data,
EOR: end-of-record; 7/8/9 multiple punch in column 1, other columns blank.

DATA DECK		SUBDECK OR CARD		
Name	Using Program	Name	Format Style	Contents and Comments
Operator	G01	Name card	1	Name/description; may be blank
		parameter card	2	user-defined parameters
		Operator subdeck	3	operator data (normally 1 card per operator and supertype definition subdecks (Note: each supertype definition must be terminated with an EOR card)
		Final signal card	3	First number on card is 0; followed by final signal numbers
		End card	EOR	
Kind	G02	name card	1X	Name/description; may be blank; a nonblank character in column 80 calls the perfect case option.
		kind subdeck	3	kind data; normally 1 card per kind
		end card	EOR	
Analysis	G03	Name card	1	name/description; may be blank
		parameter card	2	user-defined parameter
		Sensitivity subdecks (optional) for each deck:		
		name card	1X	Name/description; nonblank character in column 80 calls for parameter changes
		parameter change card	2	redefines parameters; present if and only if column 80 on previous card is nonblank.
		new kind data	3	New kind data record(s); this data replaces corresponding kind file data for this sensitivity run only (no carry over)
		end card	EOR	
		end card	EOR	

E4 Parameter Summaries

Each parameter card must be punched with the following format:

```
^$PARAM^ pname=value,...,pname=value $
```

where ^ = blank

pname = a parameter name (spelling must be correct)

value = user-selected values for the parameter.

Note: Items must be separated by a comma (and optional blanks)

If the default values are to be used for all of the parameters, the parameter card is still required and will contain:

```
^$PARAM^ $
```

GO1 PARAMETERS

Parameter Name	Function	Possible Values	Default
VALUES	Number of signal values (1)	2 to 128	200 (2)
OPS	Print operator data	0 means no 1 means yes	1 (yes)
SIGNALS	Print signal table	0 means no 1 means yes	1 (yes)
ERRORS	number of errors before aborting the run	any positive integer	25
BIAS	automatic initial bias for super operators	0 to 9999	5000
INFIN	value of infinity (3)	1 to 127	-

(1) "infinity" will be equal to this number minus one.

(2) This default value will produce an error but will permit the initial processing of the operator deck in order to check for other errors.

(3) This parameter may be used in place of VALUES; if it is specified, VALUES will be set to INFIN + 1.

GO3 PARAMETERS

Parameter Name	Function	Possible Value	Default
PMIN	Probability cut-off for pruning	0.0 to 1.0	1.E-8
NEW	Omit regular run (get new data immediately)	0 means no 1 means yes	0 (no)
INTER	Print intermediate details	0 means no 1 means yes	0 (no)
SAVE	Save all distri-butions in the distribution file	0 means no 1 means yes	0 (no)
FIRST	Number of first operator for tracing	0-10000	10000
LAST	Number of last operator for tracing	0-10000	10000
TRACE	Cut-off probability for tracing	$\geq$ 0.0	2.